

Minimum Equivalent Precedence Relation Systems

Kin Cheong Sou

March 17, 2015

Abstract

In this paper two related simplification problems for systems of linear inequalities describing precedence relation systems are considered. Given a precedence relation system, the first problem seeks a minimum subset of the precedence relations (i.e., inequalities) which has the same solution set as that of the original system. The second problem is the same as the first one except that the “subset restriction” in the first problem is removed. This paper establishes that the first problem is NP-hard. However, a sufficient condition is provided under which the first problem is solvable in polynomial-time. In addition, a decomposition of the first problem into independent tractable and intractable subproblems is derived. The second problem is shown to be solvable in polynomial-time, with a full parameterization of all solutions described. The results in this paper generalize those in [Moyle and Thompson 1969, Aho, Garey, and Ullman 1972] for the minimum equivalent graph problem and transitive reduction problem, which are applicable to unweighted directed graphs.

1 Introduction

1.1 Statement of problem

In this paper we consider precedence relation systems with n variables of the form:

$$x_i - x_j \leq c_{ij}, \quad (i, j) \in E, \quad (1)$$

where $c_{ij} \in \mathbb{R}$ and $E \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$ are given, and x_1, x_2, \dots, x_n are the variables. E is the index set of all precedence relations in (1). Let $c(E) \in \mathbb{R}^{|E|}$ be the vector edge weights such that if the k^{th} element of E is (i, j) , then $c_k = c_{ij}$. Also, let $V = \{1, 2, \dots, n\}$ denote the set of variable indices. Then, system (1) is described by the triple $(V, E, c(E))$.

This paper considers two related problems regarding the simplification of precedence relation systems.

The first problem – maximum index set of redundant relations problem: we seek a *maximum (cardinality) index set of redundant relations*, with the definition of an index set of redundant relations given by:

Definition 1 (Index set of redundant relations). Let $(V, E, c(E))$ be a precedence relation system, then $R \subseteq E$ is called an index set of redundant relations of $(V, E, c(E))$ if

$$x \in \mathbb{R}^n \text{ satisfies } (V, E, c(E)) \iff x \text{ satisfies } (V, (E \setminus R), c(E \setminus R)). \quad (2)$$

In this paper, two precedence relation systems are equivalent (with symbol \equiv) if they have the same solution set. Therefore, condition (2) can be stated as $(V, (E \setminus R), c(E \setminus R)) \equiv (V, E, c(E))$. In its minimization form, the first problem can be posed as finding a minimum (cardinality) subset $E' \subseteq E$ (i.e., $E' = E \setminus R$) such that $(V, E', c(E')) \equiv (V, E, c(E))$.

The second problem – equivalent reduction problem: we seek an *equivalent reduction* of a given precedence relation system $(V, E, c(E))$ as follows:

Definition 2 (Equivalent reduction). Let $(V, E, c(E))$ be a precedence relation system. A precedence relation system $(V, E', c'(E'))$ with $E' \subseteq V \times V$ and $c'(E') \in \mathbb{R}^{|E'|}$ is an equivalent reduction of $(V, E, c(E))$ if it satisfies the following two conditions:

2.a $x \in \mathbb{R}^n$ satisfies $(V, E, c(E)) \iff x$ satisfies $(V, E', c'(E'))$,

2.b with respect to property 2.a, E' has the minimum cardinality.

The maximum index set of redundant relations problem (i.e., the first problem) is a restriction of the equivalent reduction problem (i.e., the second problem). In the search of the minimum equivalent system of precedence relations for the first problem (in its minimization version) we are restricted to a subset of (1), whereas in the second problem there is no such restriction. In addition, according to definition the values in c' in the second problem need not be the same as c as in the first problem. The distinction between the two problems considered in this paper is analogous to the distinction between minimum equivalent graph in [1] and transitive reduction in [2], in the setting of unweighted directed graph simplification.

1.2 Main contributions and previous works

The main contributions of this paper are as follows:

1. We derive a sufficient condition under which the maximum index set of redundant relations problem has a unique solution and is polynomial-time solvable. In addition, we show that in general the maximum index set of redundant relations problem is NP-hard.
2. We show that the maximum index set of redundant relations problem can be decomposed into a finite number of independent subproblems, one of which being polynomial-time solvable and the rest NP-hard.
3. Based on the decomposition, we provide a parameterization of all solutions to the maximum index set of redundant relations problem.
4. We provide a complete parameterization of all solutions to the equivalent reduction problem. The parameterization suggests a procedure that can find any solution to this problem in polynomial time.

In essence, the results in this paper are generalizations of those in [1, 2]. The generalization is in the sense that the results in [1, 2] pertain *unweighted* directed graphs, while the results in this paper pertain *weighted* directed graphs (the relation between weighted directed graphs and precedence inequality systems will be explained in the sequel). When $c_{ij} = 0$ for all $(i, j) \in E$, it can be shown that (2) is equivalent to the condition that unweighted directed graph (V, E) , with V and E being the node set and edge set respectively, has the same reachability as $(V, E \setminus R)$. That is, there is a walk from $i \in V$ to $j \in V$ in (V, E) if and only if there is a walk from i to j in $(V, E \setminus R)$. Thus, the minimum equivalent graph problem for *unweighted* directed graphs [1] can be reduced to the maximum index set of redundant relations problem with $c_{ij} = 0$ for all $(i, j) \in E$. An implication of the reduction is that the results in this paper can be specialized to obtain those in [1]. In addition, we establish that even in the generalized setup the complexity and decomposition results in this paper are analogous to those in [1]. However, the results in this paper and in [1] are not exactly the same – there is a difference in the equivalence classes (in the node set) defining the decompositions. Similarly, an instance of the transitive reduction problem (studied in [2]) for unweighted directed graph (V, E) can be solved as an instance of equivalent reduction problem with $(V, E, c(E))$ with $c_{ij} = 0$ for all $(i, j) \in E$. In addition, the complexity result and the parameterization of the set of all equivalent reductions provided in this paper are analogous to those of [2].

In comparison with methods to simplify general sets of linear equalities and inequalities (e.g., [3,4]), this paper proposes more specialized and time-efficient algorithms for the more restrictive setting of precedence relation inequalities in (1).

1.3 Application motivations

The precedence relation system in (1) arises in applications such as machine scheduling (e.g., [5–9]), chemical process planning (e.g., [10]), smart grid (e.g., [11,12]), parallel computing (e.g., [13]) and flexible manufacture systems (e.g., [14]). In particular, in [8,9] constraints in (1) are referred to as positive and negative time-lag constraints and generalized precedence constraints, respectively. Scheduling problems with (1) are analyzed in [8,9] and the subsequent literature.

It will be shown, for instance, that the equivalent reduction problem can be solved in $O(|V|^3)$ time. Hence, algorithms that require more than $O(|V|^3)$ time for problems involving precedence constraints in (1) can potentially benefit from the simplification results in this paper. For example, [15] considered a nonconvex resource allocation problem where the decision variables x_i 's are the start times of tasks to be scheduled. In addition, x_i 's are precedence-constrained as in (1). It was shown in [15] that the computation effort for solving the resource allocation problem using dynamic programming increases exponentially with the cardinality of the minimum feedback vertex set of the undirected version of graph (V, E) . By solving the precedence relation system simplification problems in this paper, and replacing (V, E) with an equivalent (V, E') with $|E'| < |E|$, it is possible to reduce the computation effort for solving the resource allocation problem in [15].

In [16] Coffman and Graham derive an algorithm to solve a machine scheduling problem with a special case of (1), where x_i is interpreted as the start time of task i in a n -task scheduling problem and $c_{ij} = -p_i$ with p_i being the given processing time for task i . The algorithm in [16] needs to remove all redundant constraints in the specialized version of (1). This can be achieved by applying transitive reduction in [2] to an appropriately constructed unweighted directed acyclic graph. However, in [8,9] machine scheduling problems with (1) in its full generality are considered. We are not aware of any solution algorithm for the generalized machine scheduling problem that is similar to the one by Coffman and Graham in [16] using transitive reduction. However, it seems plausible that if a Coffman-Graham-like algorithm is to be developed for the general machine scheduling problem, the simplification results presented in this paper for (1) in its full generality

would be needed.

1.4 Organization

The rest of the paper is organized as follows: Section 2 defines the precedence graph associated with (1) and lists some results that are useful in the sequel. Section 2 also establishes the equivalence between the algebraic notion of index set of redundant relations with a precedence graph-based concept to be defined as redundant edge set. Then, Section 3 develops the complexity and decomposition results for the problem of finding the maximum redundant edge set. After that, Section 4 is dedicated to the problem of equivalent reduction. It characterizes the set of all equivalent reductions of any precedence relation system, and establishes that any equivalent reduction can be found in polynomial-time. Section 5 concludes the paper.

2 Graph interpretation of the main problems

Section 2.1 defines precedence graph and establishes some properties necessary for the subsequent analyses. Then, Section 2.2 describes a reformulation of the maximum index set of redundant relations problem into an equivalent graph-based problem involving the precedence graph of (1). The graph-based formulation will be studied in detail in Section 3.

2.1 Precedence graph description and supplementary results

In this paper, we make extensive use of precedence graph to describe a precedence relation system. In fact, we use these two concepts interchangeably. A precedence graph $(V, E, c(E))$ is an edge weighted directed graph with node set, edge set and vector of edge weights being V , E and $c(E)$ respectively. In addition, a precedence graph corresponds to a precedence relation system sharing the same triple $(V, E, c(E))$. The correspondence is as follows:

	precedence graph	precedence relation system
V	node set	index set of variables
E	edge set	index set of inequalities
c_{ij} for $(i, j) \in E$	edge weight	$x_i - x_j \leq c_{ij}$

The following standard graph notions are defined in order to make the subsequent discussions more precise. A walk from $u \in V$ and $v \in V$ is

defined as $(u = i_0, i_1, \dots, i_m = v)$ where $(i_k, i_{k+1}) \in E$ for $k = 0, \dots, m-1$ and the traversed nodes i_0, \dots, i_m are not necessarily distinct. A closed walk is a walk (i_0, i_1, \dots, i_m) with the additional requirement that $i_0 = i_m$. A (simple) path from u to v is a walk with the additional requirement that all traversed nodes are distinct. A cycle is a closed walk where $i_0 = i_m$ and all other traversed nodes are distinct. The weight of a walk (e.g., path, cycle) is the sum of the weights of all traversed edges, with the edge weight added as many times as an edge is traversed. This paper allows degenerate path (u) for $u \in V$ (i.e., a single node), and the weight associated with (u) is zero.

The following symbols will be used throughout the paper. For any two nodes u and v , the symbol p_{uv} is used to denote a path from u to v . The weight of a path p_{uv} is denoted $c_{p_{uv}}$. Similarly, the symbol w_{uv} is used to denote a walk from u to v with the corresponding walk weight denoted $c_{w_{uv}}$. The symbol $u \rightsquigarrow v$ is used to substitute the phrase “from u to v ”. The symbol $u \rightarrow v$ is used to denote an edge from u to v .

Because of the associated precedence inequalities, a precedence graph has the following property:

Lemma 1. *Let $G = (V, E, c(E))$ be a precedence graph. If the corresponding precedence inequality system has at least one solution, then the weights of all closed walks (e.g., cycles) in G are nonnegative.*

Proof. Let (i_0, i_1, \dots, i_m) with $i_0 = i_m$ denote a closed walk in G . By the statement assumption, there exists $x \in \mathbb{R}^n$ satisfying the inequalities corresponding to the edges in the closed walk:

$$\begin{aligned} x_{i_0} - x_{i_1} &\leq c_{i_0 i_1} \\ x_{i_1} - x_{i_2} &\leq c_{i_1 i_2} \\ &\vdots \\ x_{i_{m-1}} - x_{i_m} &\leq c_{i_{m-1} i_m}. \end{aligned} \tag{3}$$

Summing up all inequalities, with the fact that $i_0 = i_m$, leads to the desired inequality that $0 \leq c_{i_0 i_1} + c_{i_1 i_2} + \dots + c_{i_{m-1} i_m}$. \square

Certain assumptions on the graphs considered in this paper are made:

- We call a precedence graph feasible if its corresponding precedence relation system has at least one solution. In this paper, all except one precedence graphs are assumed to be feasible (the only exception is in the last part of the proof of Theorem 4). This is due to the fact that all given precedence graphs can be assumed to be feasible, and

there is no need to consider derived precedence graphs that are not feasible except in the only exception mentioned above. Hence, if there is no mentioning of the feasibility of a precedence graph, Lemma 1 is assumed to be applicable to this graph.

- It is assumed that in all graphs there is at most one (directed) edge from one node to another. In other words, no parallel edges are allowed. This assumption is obvious for precedence graphs: if for any pair (i, j) multiple precedence relations hold with $x_i - x_j \leq c_{ij}^k$ for $k = 1, 2, \dots, m$, then $x_i - x_j \leq \min_k \{c_{ij}^k\}$ summarizes the same relations. Since all other graphs (e.g., those in statement assumptions) in fact represent precedence graphs, the no-parallel-edge assumption is imposed on these graphs as well.
- It is assumed that there is no self-loop of the form (i, i) in all graphs. Again, this assumption is obvious for precedence graphs: if the precedence graph is feasible then a self-loop means $x_i - x_i = 0 \leq c_{ii}$. This inequality on c_{ii} can be removed without affecting the rest of the precedence relation system. It will also be obvious that for the only exception precedence graph whose feasibility cannot be taken for granted, there is no need to include self-loop in it.

In summary, the following are the standing assumptions of this paper:

Assumption 1.

- 1.a With only one exception in the proof of Theorem 4, all precedence graphs are feasible.
- 1.b With only one exception in the proof of Theorem 4, no precedence graph contains any negative weight closed walks.
- 1.c There is no parallel edges between nodes in any graph.
- 1.d There is no self-loop in any graph.

Remark 1. If a graph contains a negative weight cycle then it contains a negative weight closed walk. Conversely, if a graph contains a negative weight closed walk, then it must contain a negative weight cycle because a closed walk can be decomposed into a finite number of cycles, with the walk weight being the sum of the weights of the cycles (see Appendix A for further details). Therefore, Assumption 1.b is in fact equivalent to the assumption that the no precedence graph contains any negative weight cycles. In

addition, the no-negative-weight-cycle and no-negative-weight-closed-walk assumptions will be used interchangeably in this paper for convenience.

The following statement is important in the subsequent developments. For instance, the statement establishes the equivalence between the algebraic conditions in (2) and a graph theoretic condition in the corresponding precedence graph. A similar algebraic/graph equivalence can be established for the case of Definition 2.a, with the aid of this same statement.

Lemma 2. *Let $G = (V, E, c(E))$ be a precedence graph, and its corresponding precedence inequality system be*

$$x_i - x_j \leq c_{ij}, \quad (i, j) \in E. \quad (4)$$

Let $u, v \in V$, $u \neq v$, $b_{uv} \in \mathbb{R}$ be given. Then the following two conditions are equivalent:

2.a Whenever $x \in \mathbb{R}^n$ satisfies (4), x satisfies the inequality $x_u - x_v \leq b_{uv}$.

2.b There exists a path p_{uv} , $u \rightsquigarrow v$ in G such that the weight of p_{uv} , denoted $c_{p_{uv}}$, satisfies $c_{p_{uv}} \leq b_{uv}$.

Proof. Since $u \neq v$, we can define $a_{uv} \in \mathbb{R}^n$ such that

$$a_{uv}(k) = \begin{cases} 1 & k = u \\ -1 & k = v \\ 0 & \text{otherwise} \end{cases}.$$

With (standing) Assumption 1.d, if $(i, j) \in E$ then $i \neq j$. Hence, for each $(i, j) \in E$ we can define the vector $a_{ij} \in \mathbb{R}^n$ such that

$$a_{ij}(k) = \begin{cases} 1 & k = i \\ -1 & k = j \\ 0 & \text{otherwise} \end{cases}$$

That is, a_{ij} defines the column of the incidence matrix of G for edge (i, j) . With a_{ij} , the inequality $x_i - x_j \leq c_{ij}$ can be written as $a_{ij}^T x \leq c_{ij}$. Furthermore, let J_P denote the optimal objective value of the following linear program

$$\begin{aligned} J_P := & \max_{x \in \mathbb{R}^n} a_{uv}^T x \\ \text{subject to} & a_{ij}^T x \leq c_{ij}, \quad \forall (i, j) \in E. \end{aligned} \quad (5)$$

Then, condition 2.a holds if and only if $J_P \leq b_{uv}$. The linear programming dual of (5) and its optimal objective value can be written as

$$\begin{aligned} J_D := & \min_{y_{ij} \in \mathbb{R}} \sum_{(i,j) \in E} y_{ij} c_{ij} \\ \text{subject to} & \sum_{(i,j) \in E} y_{ij} a_{ij} = a_{uv} \\ & y_{ij} \geq 0, \quad \forall (i,j) \in E. \end{aligned} \quad (6)$$

Due to Assumption 1.a, (4) has at least one solution. Hence, the feasible set of (5) is nonempty. Thus, by linear programming duality (e.g., [17]) it holds that $J_D = J_P$ with the convention that $J_D = \infty$ whenever (6) is infeasible. Hence,

$$\text{condition 2.a holds} \iff J_P \leq b_{uv} \iff J_D \leq b_{uv}. \quad (7)$$

Note that a_{uv} contains only -1 , 1 or 0 . In addition, a_{ij} for $(i,j) \in E$ in the constraint of (6) are columns of an incidence matrix (of G), which is totally unimodular (e.g., [18]). Thus, a standard combinatorial optimization argument (e.g., [18]) implies that the relations in (7) can be extended to

$$\text{condition 2.a holds} \iff J_P \leq b_{uv} \iff J_D \leq b_{uv} \iff J_B \leq b_{uv}, \quad (8)$$

where J_B is the optimal objective value of the following 0 – 1 binary linear integer problem

$$\begin{aligned} J_B := & \min_{y_{ij} \in \mathbb{R}} \sum_{(i,j) \in E} y_{ij} c_{ij} \\ \text{subject to} & \sum_{(i,j) \in E} y_{ij} a_{ij} = a_{uv} \\ & y_{ij} \in \{0, 1\}, \quad \forall (i,j) \in E. \end{aligned} \quad (9)$$

That is, (9) is almost the same as (6) except that the decision variables in (9) are restricted to 0 or 1. Next, we establish that $J_B \leq b_{uv}$ is equivalent to condition 2.b. That is,

$$J_B \leq b_{uv} \iff \exists \text{ path } p_{uv} : u \rightsquigarrow v \text{ in } G \text{ such that } c_{p_{uv}} \leq b_{uv}. \quad (10)$$

One side of the implication is easy to establish: if there is a path $p_{uv} : u \rightsquigarrow v$ in G such that $c_{p_{uv}} \leq b_{uv}$, then by assigning $y_{ij} = 1$ if and only if (i,j) is part of the path we obtain a feasible solution of (9) with objective value being $c_{p_{uv}} \leq b_{uv}$. Hence, $J_B \leq b_{uv}$. Conversely, if $J_B \leq b_{uv}$ then the following procedure can be employed to retrieve a walk $p_{uv} : u \rightsquigarrow v$ in G such that $c_{p_{uv}} \leq b_{uv}$:

- Initialize $W \leftarrow \emptyset$, and $Y \leftarrow \{(i, j) \in E \mid y_{ij}^* = 1\}$ with y^* being an optimal solution to (9)
- While $Y \neq \emptyset$, do
 1. If $(u, s) \in Y$ for some s then set $(i, j) \leftarrow (u, s)$, otherwise set (i, j) as an arbitrary edge in Y
 2. $Y \leftarrow Y \setminus \{(i, j)\}$
 3. $w \leftarrow (i, j)$
 4. While $(j, t) \in Y$ for some t , do
 - (a) $Y \leftarrow Y \setminus \{(j, t)\}$
 - (b) $w \leftarrow w + t$ (meaning that w is appended in the end by t)
 - (c) $j \leftarrow t$
 5. End (of second while)
 6. $W \leftarrow W \cup \{w\}$
- End (of first while)

To analyze the procedure the following nonnegative degree counters are needed: $d^+(i)$ is the out-degree at node i in the graph (V, Y) , where Y is being updated in the procedure. Similarly, $d^-(i)$ is the in-degree at node i . Initially, the degree counters satisfy

$$d^+(u) = d^-(u) + 1, \quad d^+(v) = d^-(v) - 1, \quad d^+(i) = d^-(i), \quad i \neq u, v, \quad (11)$$

because for each $k \in \{1, 2, \dots, n\}$ the constraint in (9) can be written as

$$\sum_{(i,j) \in E} y_{ij}^* a_{ij}(k) = \sum_{j \mid (k,j) \in E} y_{kj}^* - \sum_{j \mid (j,k) \in E} y_{jk}^* = a_{uv}(k) = \begin{cases} 1 & k = u \\ -1 & k = v \\ 0 & k \neq u, v \end{cases}.$$

Now we first analyze some general properties of the procedure. Notice that the procedure should terminate in finite number of passes of the first while-loop because within each pass at least one edge is removed from Y and Y is finite. Also, with a similar argument we can establish that the second while-loop terminates for each pass of the first while-loop. At the end of a pass of the first while-loop, a walk $w = (i = i_0, i_1, \dots, i_m)$ is retrieved. There can be two possibilities with i_m : either (I) $i_m = i_0$ (i.e., w is a closed walk) or (II) $i_m \neq i_0$. Let $d_{\text{fore}}^+(i)$ and $d_{\text{fore}}^-(i)$ denote, respectively, the out-degree and in-degree at node i before the pass of the first while-loop. Similarly,

let $d_{\text{aft}}^+(i)$ and $d_{\text{aft}}^-(i)$ denote the analogous degrees after the pass of the first while-loop. Then, for case (I) with $i_0 = i_m$ the degree adjustments (after a pass of the first while-loop) are as follows:

$$\begin{aligned} d_{\text{aft}}^+(i_q) &= d_{\text{fore}}^+(i_q) - r_q, \quad \forall q \in \{0, 1, \dots, m\}, \\ d_{\text{aft}}^-(i_q) &= d_{\text{fore}}^-(i_q) - r_q, \quad \forall q \in \{0, 1, \dots, m\}, \\ d_{\text{aft}}^+(i_m) &= 0. \end{aligned} \tag{12}$$

In (12), r_q are some nonnegative numbers satisfying $d_{\text{aft}}^+(i_q) \geq 0$ and $d_{\text{aft}}^-(i_q) \geq 0$. In other words, the difference between the out-degree and in-degree for each node remains the same in case (I) (when w is a closed walk). For case (II) with $i_0 \neq i_m$ the degree adjustments are as follows:

$$\begin{aligned} d_{\text{aft}}^+(i_0) &= d_{\text{fore}}^+(i_0) - r_0, \\ d_{\text{aft}}^-(i_0) &= d_{\text{fore}}^-(i_0) - r_0 + 1, \\ d_{\text{aft}}^+(i_q) &= d_{\text{fore}}^+(i_q) - r_q, \quad \forall q \in \{1, 2, \dots, m-1\}, \\ d_{\text{aft}}^-(i_q) &= d_{\text{fore}}^-(i_q) - r_q, \quad \forall q \in \{1, 2, \dots, m-1\}, \\ d_{\text{aft}}^+(i_m) &= d_{\text{fore}}^+(i_m) - r_m + 1, \\ d_{\text{aft}}^-(i_m) &= d_{\text{fore}}^-(i_m) - r_m, \\ d_{\text{aft}}^+(i_m) &= 0. \end{aligned} \tag{13}$$

From (13), the in-degree counter of the end node i_m must satisfy

$$d_{\text{aft}}^-(i_m) = d_{\text{fore}}^-(i_m) - d_{\text{fore}}^+(i_m) - 1. \tag{14}$$

Now we analyze the first pass of the first while-loop. With $J_B \leq b_{uv}$ assumed (hence feasibility), the constraint in (9) requires that $(u, s) \in Y$ for some s . Hence, $i_0 = u$ for the walk w in the first pass. In addition, w cannot be a closed walk because (11) (with degree counters interpreted as those before the pass) and (12) together imply that $d_{\text{aft}}^-(u) = d_{\text{fore}}^-(u) - d_{\text{fore}}^+(u) = -1$ which is impossible. Hence, case (II) must hold. Then, (11) (with degree counters interpreted as those before the pass) and (14) together imply that

$$i_m = v, \quad d_{\text{aft}}^-(v) = 0, \tag{15}$$

since all other choices of i_m would result in negative in-degree after the pass. Therefore, w is in fact a walk from u to v , which is denoted as w_{uv} and stored in W in step 6. Furthermore, (13) and (15) together suggest that after the

first pass of the first while-loop the degree counters (in particular those of u) are updated to

$$d^+(g) = d^-(g), \quad \forall g \in V, \quad d^+(v) = d^-(v) = 0. \quad (16)$$

In summary, v is no longer incident to any edge in Y and the out-degree and in-degree for each node become equal (including the case of u), when the first pass of the first while-loop is finished.

Next, we analyze the subsequent passes of the first while-loop. (16) (with degree counters interpreted as those before the second pass) and (14) implies that if the walk in the second pass is not a closed walk then v must be the end node of this walk. Therefore, the walk from the second pass must be a close walk denoted w_2 because (16) specifies that v is no longer incident to any edge in Y at this stage. In addition, by (12) the in/out degree difference for each node remains zero after the second pass. This pattern continues for all subsequent passes, generating closed walks w_3, \dots, w_N until Y becomes an empty set. Upon termination of the first while-loop, W contains w_{uv} (i.e., the walk from u to v) and closed walks w_2, w_3, \dots, w_N . In addition, it can be seen that

$$c_{w_{uv}} + c_{w_2} + c_{w_3} + \dots + c_{w_N} = J_B \leq b_{uv}.$$

This suggests that $c_{w_{uv}} \leq b_{uv}$ since Assumption 1.b states that $c_{w_q} \geq 0$ for $q = 2, 3, \dots, N$. The walk w_{uv} may not be a path from u to v . However, as explained in Appendix A w_{uv} can be further decomposed into a path $p_{uv} : u \rightsquigarrow v$ and a finite number of cycles, which have nonnegative weights because of Assumption 1.b (this argument will in fact be formalized in Lemma 3 to be described). Hence, $p_{uv} \leq w_{uv} \leq b_{uv}$. In conclusion, when $J_B \leq b_{uv}$, we can indeed obtain a path $p_{uv} : u \rightsquigarrow v$ with weight $p_{uv} \leq b_{uv}$. This establishes (10). Finally, combining (8) and (10) yields the desired statement that

$$\text{condition 2.a holds} \iff J_B \leq b_{uv} \iff \text{condition 2.b holds.}$$

□

2.2 Graph representation of maximum index set of redundant relations problem

For the analysis in the sequel, the notion of index set of redundant relations defined in (2) will be reinterpreted as an equivalent and graph based concept of *redundant edge set* to be defined in Definition 3. Consequently, the problem of finding the maximum index set of redundant relations in (1) can

be reformulated as the problem of finding the *maximum redundant edge set* in the corresponding precedence graph. A consequence of the reformulation, as will be clear, is that the algebraic problem of finding the maximum index set of redundant relations can be solved using readily available graph-based algorithms (e.g., Floyd-Warshall shortest path algorithm).

Definition 3 (Redundant edge set). For an edge weighted directed graph $G = (V, E, c(E))$, an edge subset $R \subseteq E$ is called a redundant edge set of G if either $R = \emptyset$ or when $R \neq \emptyset$, it holds that

$$\begin{aligned} &\text{for every } (u, v) \in R \text{ with weight } c_{uv}, \text{ there exists a path } p_{uv} \\ &\text{in } (V, E \setminus R, c(E \setminus R)) \text{ from } u \text{ to } v \text{ such that } c_{p_{uv}} \leq c_{uv}. \end{aligned} \quad (17)$$

Remark 2. For an edge weighted directed graph without negative weight cycles (e.g., a precedence graph), the definition of redundant edge set can be relaxed. Specifically, the existence of a path p_{uv} in condition (17) can be replaced with the existence of a walk w_{uv} from u to v in $(V, E \setminus R, c(E \setminus R))$ such that $c_{w_{uv}} \leq c_{uv}$. The relaxation is justified by the following statement.

Lemma 3. Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight cycles (e.g., a precedence graph). For any two nodes $u, v \in V$ with $u \neq v$. If $w_{uv} : (u = i_0, i_1, \dots, i_m = v)$ is a walk such that $c_{w_{uv}} \leq c_{uv}$. Then there exists a path $p_{uv} : (i_{k(1)}, i_{k(2)}, \dots, i_{k(q)})$ where $k(1), k(2), \dots, k(q)$ is a subsequence of $0, 1, \dots, m$ such that $k(1) = 0$ and $k(q) = m$. In addition, the weight of p_{uv} , denoted $c_{p_{uv}}$, satisfies $c_{p_{uv}} \leq c_{w_{uv}} \leq c_{uv}$.

Proof. According to Appendix A, the walk w_{uv} can be decomposed into a path p_{uv} of the form described in the statement and a finite number of cycles. In addition, the weight of the walk w_{uv} is the sum of the weight of the path p_{uv} and those of the cycles (if any). Since the weight of all cycles are nonnegative as assumed in the statement, it holds that $c_{p_{uv}} \leq c_{w_{uv}}$. \square

Now we establish the equivalence between the notions of index set of redundant relations and redundant edge set.

Lemma 4. Let $G = (V, E, c(E))$ denote a precedence relation system and its precedence graph. Then $R \subseteq E$ is an index set of redundant relations of G (Definition 1) if and only if it is a redundant edge set of G (Definition 3).

Proof. The statement holds trivially if $R = \emptyset$. Therefore, the rest of the proof assumes that $R \neq \emptyset$. By Definition 1, R is an index set of redundant

relations if and only if

$$\begin{aligned} & x \in \mathbb{R}^n \text{ satisfies } x_u - x_v \leq c_{uv}, \quad \forall (u, v) \in (E \setminus R) \\ \implies & x_u - x_v \leq c_{uv}, \quad \forall (u, v) \in R. \end{aligned} \tag{18}$$

$(V, E \setminus R, c(E \setminus R))$, being a subgraph of G , is a precedence graph (with Assumption 1 satisfied). By applying Lemma 2 with $(V, E \setminus R, c(E \setminus R))$ for $b_{uv} = c_{uv}$ for each $(u, v) \in R$, the condition in (18) is equivalent to

for each $(u, v) \in R$ there exists a path $p_{uv} : u \rightsquigarrow v$
in $(V, E \setminus R, c(E \setminus R))$ such that $c_{p_{uv}} \leq c_{uv}$.

This is the same as the condition that R is a redundant edge set, according to Definition 3. □

In view of Lemma 4, the problem of finding the maximum index set of redundant relations in (1) can be posed as the problem of finding the *maximum (cardinality) redundant edge set* in the precedence graph associated with (1), with the definition of redundant edge set given in (17).

3 Maximum redundant edge set problem

This section presents the results on the maximum redundant edge set problem. In Section 3.1 we will establish that the maximum redundant edge set problem is polynomial-time solvable if the precedence graph does not have any zero-weight cycle. On the other hand, we will show that in general the problem is NP-hard. Section 3.2 specifies that the maximum redundant edge set problem can be decomposed into several subproblems: one subproblem is polynomial-time solvable and the other subproblems are NP-hard. Section 3.3 discusses some computation issues pertaining the decomposition result in Section 3.2.

3.1 Complexity of maximum redundant edge set problem

This subsection answers the following basic and complexity related questions regarding the maximum redundant edge set problem.

1. When is there a nonempty redundant edge set?
2. If there exists at least one redundant edge set, when is the maximum redundant edge set unique and how to compute it?

3. When is it computationally intractable to solve the maximum redundant edge set problem?

The answer to the first question lies in the following concept:

Definition 4 (Redundant edge). For an edge weighted directed graph $G = (V, E, c(E))$, an edge $(i, j) \in E$ is called a redundant edge of G if the singleton $\{(i, j)\}$ is a redundant edge set of G (defined in (17)).

Remark 3. In general, an arbitrary set of redundant edges need not be a redundant edge set defined in (17) in Definition 4, because the “replacement path” of one redundant edge might contain another redundant edge.

Lemma 5. Let $G = (V, E, c(E))$ be an edge weighted directed graph. Then, there exists a nonempty redundant edge set in G if and only if there exists a redundant edge in G .

Proof. The sufficiency part is due to the fact that a redundant edge leads to a single-member redundant edge set. For the necessity part, note that by definition in (17) any subset of a redundant edge set is also a redundant edge set. Hence, nonexistence of redundant edge implies nonexistence of single-member and in general any redundant edge sets. \square

The answer to the second question is provided by the following statement (Lemma 6) and its corollary (Theorem 1), whose preliminary version appeared in [19]:

Lemma 6. Let $G = (V, E, c(E))$ be an edge weighted directed graph. Assume that all cycles in G have positive weights. Then, if R and R' are redundant edge sets of G satisfying (17), $R \cup R'$ is also a redundant edge set of G .

Proof. To verify that $R \cup R'$ is indeed a redundant edge set, it is sufficient to verify that for each $(u, v) \in R \cup R'$ with weight c_{uv} , it holds that

$$\begin{aligned} &\exists \text{ a path } p_{uv} \text{ in } (V, E \setminus (R \cup R'), c(E \setminus (R \cup R'))) \text{ from } u \text{ to } v \\ &\text{such that } c_{p_{uv}} \leq c_{uv}. \end{aligned} \tag{19}$$

In the subsequent parts of the proof, the following shorthand is used: the sentence “a walk w is in $\hat{E} \subseteq E$ ” means that w is in the subgraph $(V, \hat{E}, c(\hat{E}))$. Without loss of generality, assume that $(u, v) \in R$ (otherwise we exchange the roles of R and R'). Then by (17) there exists a walk (which is in fact a path) $w_1 : u \rightsquigarrow v$ in $E \setminus R$ such that $c_{w_1} \leq c_{uv}$. If w_1 is also in $E \setminus R'$ then w_1 is in $E \setminus (R \cup R')$, and hence (u, v) satisfies (19). If, on the other hand,

w_1 involves edges belonging to R' , then by (17) each of these edges can be replaced by a corresponding path in $E \setminus R'$. Also by (17), the weights of the replacement paths are no more than the weights of the corresponding edges. This results in another walk $w_2 : u \rightsquigarrow v$ in $E \setminus R'$, with possible edges in R . The walk w_2 has strictly more nodes (and edges) than w_1 , and $c_{w_2} \leq c_{w_1}$. If w_2 is in $E \setminus R$ then (u, v) satisfies (19), as argued above. Otherwise, the process of finding replacement walks w_3, w_4, \dots with increasing number of nodes and nonincreasing weights would continue. Next, we show by contradiction that the replacement-walk-finding process would terminate in a finite number of iterations. It is noted that any walk from u to v (with $u \neq v$) can be decomposed into one path from u to v and a finite number of cycles (see Appendix A for a proof). In addition, in a finite graph the numbers of possible paths and cycles are finite. Therefore, by the pigeon-hole principle, if the replacement-walk-finding process does not terminate at some iteration it will generate a walk \tilde{w} traversing a cycle, and one of the previously constructed walks, denoted \hat{w} , is exactly the same as \tilde{w} except that the cycle is not traversed. The construction of the walks specifies that $c_{\tilde{w}} \leq c_{\hat{w}}$, and this implies that the cycle has nonpositive weight. This contradicts the assumption that all cycles in G have positive weights. Therefore, the replacement-walk-finding process terminates in a finite number of iterations. Consequently, a walk w^* in $E \setminus (R \cup R')$ from u to v with $c_{w^*} \leq c_{uv}$ is resulted. Then, by Lemma 3 the desired path p_{uv} can be constructed from w^* to satisfy (19). Applying the same proof to all members of $R \cup R'$ completes the proof. \square

Theorem 1. *Let $G = (V, E, c(E))$ be an edge weighted directed graph. Assume that all cycles in G have positive weights. Then the maximum redundant edge set is unique. In addition, if there is a redundant edge, then the maximum redundant edge set is the set of all redundant edges, which can be computed in $O(|V|^3)$ time.*

Before the proof is given, the procedure to compute all redundant edges when G does not have zero or negative weight cycle is described first.

Algorithm 1 (Finding all redundant edges in graph $G = (V, E, c(E))$ without zero or negative weight cycles).

1. Solve the all-pair shortest path problem for all source/destination pairs in G . Let d_{ij} denote the shortest path distance from i to j .
2. An edge $(i, j) \in E$ is declared a redundant edge if and only if

$$\min_{(i,k) \in E, k \neq i, k \neq j} \{c_{ik} + d_{kj}\} \leq c_{ij}. \quad (20)$$

Lemma 7. When $G = (V, E, c(E))$ does not have any zero or negative weight cycles, Algorithm 1 correctly computes all redundant edges in $O(|V|^3)$ time.

Proof. According to Definition 4, edge (i, j) is a redundant edge if and only if there exists a path $p_{ij} : i \rightsquigarrow j$ in G such that (i, j) is not part of p_{ij} and $c_{p_{ij}} \leq c_{ij}$. If (20) does not hold, then except possibly (i, j) there is no path in G from i to j with weight less than or equal to c_{ij} . Hence, (i, j) cannot be a redundant edge. On the other hand, if (20) holds, then there exists a walk $w_{ikj} : i \rightarrow k \rightsquigarrow j$ such that $c_{w_{ikj}} \leq c_{ij}$. If (i, j) is part of w_{ikj} then the walk is of the form $i \rightarrow k \rightsquigarrow i \rightarrow j \rightsquigarrow j$, and its weight is $c_{w_{ikj}} > c_{ij}$ because the closed walks $i \rightarrow k \rightsquigarrow i$ and $j \rightsquigarrow j$ can be decomposed into sequences of cycles and the cycles in G are positively weighted as assumed in the statement. This is a contradiction because $c_{w_{ikj}} \leq c_{ij}$ and $c_{w_{ikj}} > c_{ij}$ cannot be both true. Hence, (i, j) is not part of w_{ikj} . From Lemma 3 the desired path $p_{ij} : i \rightarrow k \rightsquigarrow j$ can be found to certify that (i, j) is indeed a redundant edge.

Next we argue for the computation requirement. Since G does not have any negative weight cycle. The all-pair shortest path problem can be solved using, for instance, the Floyd-Warshall algorithm in $O(|V|^3)$ time (e.g., [20]). The third step requires $O(|E||V|) = O(|V|^3)$ computation cost. \square

Remark 4. In Lemma 7 the assumption of no zero or negative weight cycles cannot be removed. See Figure 1 for a consequence of when the assumption is not satisfied.

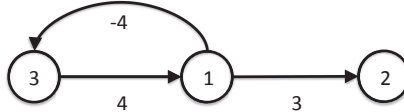


Figure 1: A 3-node example of a graph with a zero weight cycle $(1, 3, 1)$. In this example, $c_{12} = 3$, $c_{13} = -4$ and $d_{32} = 7$. In (20) for edge $(1, 2)$, $c_{13} + d_{32} = 3 \leq c_{12}$. However, $(1, 2)$ is not a redundant edge.

Proof of Theorem 1. Suppose R and R' are two different maximum redundant edge sets. Then, $|R \cup R'| > |R|$. Further, by Lemma 6 $R \cup R'$ is a redundant edge set which has more edges than R , contradicting the assumption that R is a maximum redundant edge set. Thus, the maximum redundant edge set is unique. It is denoted as R^* .

Next, let U denote the set of all redundant edges. Under the additional assumption that $U \neq \emptyset$, Lemma 5 specifies that $R^* \neq \emptyset$. Then, each $(u, v) \in R^*$ is a redundant edge because the singleton $\{(u, v)\} \subseteq R^*$ is a redundant edge set. Therefore, $R^* \subseteq U$. On the other hand, Lemma 6 states that U , which is the union of single-member redundant edge sets, is also a redundant edge set. Thus, $U \subseteq R^*$. Finally, the claim about $O(|V|^3)$ computation time follows from Lemma 7. \square

The positive weight cycle assumption in Lemma 6 is necessary. The presence of zero or negative weight cycles can indeed results in situations where the union of two redundant edge sets is not a redundant edge set, and as a result the maximum redundant edge set is not unique. For a counterexample, consider the graph in Figure 2.

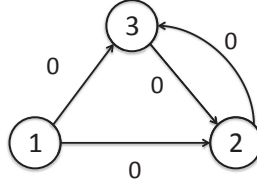


Figure 2: A 3-node example of a graph with zero weight cycles. There are two different maximum redundant edge sets, namely $\{(1, 2)\}$ and $\{(1, 3)\}$. However, the union $\{(1, 2), (1, 3)\}$ is not a redundant edge set according to (17). The cycle $(2, 3, 2)$ has zero weight.

In fact, the general maximum redundant edge set problem (without the assumption in Lemma 6) is NP-hard. This can be shown by a reduction from the NP-hard minimum equivalent graph problem studied in [1]. The proof also establishes the connection that the maximum redundant edge set problem is a generalization of the minimum equivalent graph problem.

Theorem 2. *Let G be an edge weighted directed graph. The problem of finding the maximum redundant edge set of G is NP-hard.*

Proof. First, the minimum equivalent graph problem in [1] is summarized, in the context of our discussion. Consider a directed graph (V, E) , and let $\hat{R} \subseteq E$. We mention that (V, E) and $(V, E \setminus \hat{R})$ have the same reachability if the following condition is satisfied: there is a walk (hence a path) from $i \in V$ to $j \in V$ in (V, E) if and only if there is a walk from i to j in $(V, E \setminus \hat{R})$. Then, an instance of the minimum equivalent graph problem (associated

with (V, E)) seeks the maximum cardinality $R^* \subseteq E$ such that (V, E) and $(V, E \setminus R^*)$ have the same reachability.

Next, we show that every instance of the minimum equivalent graph problem can be reduced into an instance of the maximum redundant edge set problem. To begin, we claim that

$$(V, E) \text{ and } (V, E \setminus \hat{R}) \text{ have the same reachability.} \quad (21a)$$

$$\iff \forall (i, j) \in \hat{R}, \text{ there exists a walk from } i \text{ to } j \text{ in } (V, E \setminus \hat{R}). \quad (21b)$$

Since a walk in $(V, E \setminus \hat{R})$ is a walk in (V, E) , condition (21a) is the same as the condition that

$$\exists \text{ walk } w_{ij} : i \rightsquigarrow j \text{ in } (V, E) \implies \exists \text{ walk } w_{ij}^r : i \rightsquigarrow j \text{ in } (V, E \setminus \hat{R}). \quad (22)$$

If (21b) holds, then in (22) for every edge $(u, v) \in \hat{R}$ that is part of w_{ij} there is a walk $w_{uv}^r : u \rightsquigarrow v$ in $(V, E \setminus \hat{R})$. Hence, (22) holds. On the other hand, suppose (21b) does not hold, and let $(s, t) \in \hat{R}$ be an edge such that there is no walk from s to t in $(V, E \setminus \hat{R})$. Then, with $i = s$, $j = t$ and $w_{ij} = (s, t)$ as a counterexample it can be seen that (22) does not hold. Therefore, (21a), (21b) and (22) are all equivalent. Further, we define $(V, E, c^0(E))$ where $c_{ij}^0 = 0$ for all $(i, j) \in E$ (in fact, the following argument would hold as long as $c_{ij}^0 = \alpha$ for $\alpha \leq 0$). Since any walk in (V, E) is a zero weight walk in $(V, E, c^0(E))$ and vice versa, it can be seen that (21b) is equivalent to the condition that \hat{R} is a redundant edge set in $(V, E, c^0(E))$. This suggests that an instance of minimum equivalent graph problem with (V, E) is equivalent to the instance of maximum redundant edge set problem with $(V, E, c^0(E))$, with the two problem instances having the same optimal solutions.

To complete the complexity argument, note that by definition of $c^0(E)$, (V, E) contains a cycle if and only if $(V, E, c^0(E))$ contains a zero weight cycle. Thus, if there would be a polynomial time algorithm which can solve all instances of the maximum redundant edge set problem including those with zero or negative weight cycles, then the minimum equivalent graph problem could be solved in polynomial time as well. However, since in general the minimum equivalent graph problem is NP-hard (e.g., [21]), we establish that in general the maximum redundant edge set problem is NP-hard as well. \square

In summary, if G is a precedence graph then by standing Assumption 1.b G does not have any negative weight cycles. If in addition G does not have any zero-weight cycles, then Theorem 1 states that the maximum redundant

edge set of G is unique, and it is the set of all redundant edges (if the set is nonempty). In addition, finding all redundant edges using Algorithm 1 requires $O(|V|^3)$ time. On the other hand, in the more general case where G is allowed to have zero-weight cycles, the example in Figure 2 indicates that the maximum redundant edge set need not be unique. In addition, Theorem 2 states that the maximum redundant edge set problem is NP-hard in general.

It turns out that the maximum redundant edge set problem can always be decomposed into a finite number of decoupled subproblems, one of which is solvable in polynomial time and all other are NP-hard. This decomposition, which will be detailed in Section 3.2, is analogous to that in [1] for the minimum equivalent graph problem for unweighted directed graphs.

3.2 Decomposition of maximum redundant edge set problem

In this subsection, we first introduce an equivalence class partitioning of the node set of a precedence graph, and define an auxiliary graph induced by the equivalence class partitioning called condensation. Next, we present some properties of the equivalence classes and the condensation. After that, we establish the fact that the maximum redundant edge set problem can be decomposed into $K + 1$ independent subproblems, where K is the number of equivalence classes. The main result will be summarized in Theorem 3.

In any edge weighted directed graph, we define an equivalence relation on the node set as follows:

Definition 5 (Equivalence relation \sim). Let $G = (V, E, c(E))$ denote an edge weighted directed graph. For any pair $i \in V, j \in V$, we denote $i \sim j$ if either (a) $i = j$, or (b) there exists a zero-weight closed walk in G traversing both i and j .

Remark 5. To verify that \sim is indeed an equivalence relation, it suffices to note that if $i \sim j$ and $j \sim k$ then there exist two zero-weight closed walks $i \rightsquigarrow j \rightsquigarrow i$ and $j \rightsquigarrow k \rightsquigarrow j$. Consequently, a zero-weight closed walk $i \rightsquigarrow j \rightsquigarrow k \rightsquigarrow j \rightsquigarrow i$ exists, and this implies that $i \sim k$.

The relation \sim defines equivalence classes in the node set. For convenience, we will define some notations associated with the equivalence classes. However, before these notations are defined, the notion of the minimum walk weight should be defined first.

Definition 6 (Minimum walk weight). Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. For $i \in V$,

$j \in V$, define d_{ij} to be the minimum weight of the walk among all walks in G which goes from i to j . Note that $d_{ii} = 0$ for all $i \in V$, and this is attained by the single-node path (i) since G does not have any negative weight closed walks.

Definition 7 (Equivalence classes induced by equivalence relation \sim). Let $G = (V, E, c(E))$ denote an edge weighted directed graph without negative weight closed walks. Let relation \sim be defined in Definition 5. In addition, let d_{ij} , the minimum walk weight in G , be defined in Definition 6. We define the following:

- 7.a K denotes the number of equivalence classes in V defined by relation \sim .
- 7.b For $k \in \{1, 2, \dots, K\}$, $[v_k] \subseteq V$ denotes the equivalence class containing v_k , where v_k is the (arbitrarily) designated representing node for equivalence class containing v_k .
- 7.c For $k \in \{1, 2, \dots, K\}$, we denote
 - $E_k := \{(i, j) \in E \mid i \in [v_k], j \in [v_k]\}$. That is, E_k denotes the set of edges connecting two nodes inside an equivalence class $[v_k]$.
 - Let $E_k^r \subseteq E_k$ be defined as

$$E_k^r := \{(i, j) \in E_k \mid c_{ij} > d_{ij}\}. \quad (23)$$

That is, $E_k^r \subseteq E_k$ is a subset of E_k where each member edge has an edge weight strictly greater than the corresponding minimum walk weight. As it will become apparent in the sequel, edges in E_k^r are redundant (i.e., can always be included in any maximum redundant edge set). This motivates the use of superscript “r” in (23).

- Let $E_k^c \subseteq E_k$ be defined as

$$E_k^c := \{(i, j) \in E_k \mid c_{ij} = d_{ij}\}. \quad (24)$$

That is, $E_k^c = E_k \setminus E_k^r$ since $c_{ij} \geq d_{ij}$ for all $(i, j) \in E$. This motivates the superscript “c”, since E_k^c is the complement of E_k^r .

- 7.d For $i \in \{1, 2, \dots, K\}, j \in \{1, 2, \dots, K\}, i \neq j$ we denote
 - $E_{ij} := \{(u, v) \in E \mid u \in [v_i], v \in [v_j]\}$. That is, E_{ij} denotes the set of edges from a node in $[v_i]$ to another node in $[v_j]$ with $[v_i] \neq [v_j]$ (because of the assumption that $i \neq j$).

- Let $E_{ij}^c \subseteq E_{ij}$ be defined as

$$E_{ij}^c := \{(u, v) \in E_{ij} \mid (u, v) \in \operatorname{argmin}_{(s,t) \in E_{ij}} d_{v_i s} + c_{st} + d_{t v_j}\}. \quad (25)$$

While in (25) the definition of E_{ij}^c assumes a designation of the representing nodes v_i and v_j , it turns out that E_{ij}^c is in fact independent of the choice of the designation. This will be argued in Remark 8.

- For $E_{ij} \neq \emptyset$ (hence $E_{ij}^c \neq \emptyset$), we (arbitrarily) designate a particular edge $(v_{ij}^i, v_{ij}^j) \in E_{ij}^c$ (with $v_{ij}^i \in [v_i]$, $v_{ij}^j \in [v_j]$) as the “representing” edge for E_{ij} .

7.e Collecting all inter-equivalence class edges, we denote

$$\begin{aligned} E_0 &:= \bigcup_{1 \leq i \neq j \leq K} E_{ij} \\ E_0^c &:= \bigcup_{1 \leq i \neq j \leq K} E_{ij}^c, \\ E_0^d &:= \bigcup_{1 \leq i \neq j \leq K} \{(v_{ij}^i, v_{ij}^j)\} \end{aligned} \quad (26)$$

where $1 \leq i \neq j \leq K$ is shorthand for $\{(i, j) \mid 1 \leq i, j \leq K, i \neq j\}$. It holds that $E_0^d \subseteq E_0^c \subseteq E_0$.

Figure 3 shows an illustration of the equivalence classes defined by relation \sim . Analogous to the condensation of a unweighted graph, we define the condensation of an edge weighted directed graph as follows:

Definition 8 (Condensation). Let $G = (V, E, c(E))$ denote an edge weighted directed graph without negative weight closed walks. Let other involved symbols be defined in Definitions 5, 6, 7 in the context of G . We define the condensation of G , denoted $\tilde{G} := (\tilde{V}, \tilde{E}_0, \tilde{c}(\tilde{E}_0))$ as follows:

- The set of all nodes of \tilde{G} is $\tilde{V} := \{v_1, v_2, \dots, v_K\}$.
- In \tilde{G} , there is an edge $(v_i, v_j) \in \tilde{E}_0$ with $i \neq j$ if and only if in G the set $E_{ij} \neq \emptyset$ (see Definition 7.d for E_{ij}).
- For any $(v_i, v_j) \in \tilde{E}_0$, the edge weight $\tilde{c}_{v_i v_j}$ is defined as

$$\tilde{c}_{v_i v_j} := \min_{(u,v) \in E_{ij}} d_{v_i u} + c_{uv} + d_{v v_j}, \quad (27)$$

where $d_{v_i u}$ and $d_{v v_j}$ are defined in Definition 6.

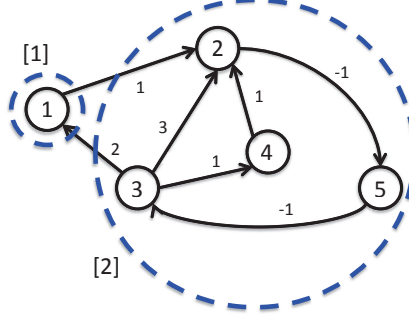


Figure 3: An example precedence graph with five nodes. The number on each edge denotes the weight of the corresponding edge. Node 1 defines an equivalence class [1] by itself because it is not on any zero weight closed walk. $\{2, 3, 4, 5\}$ is the other equivalence class [2] because of the zero-weight cycle (i.e., closed walk) $(3, 4, 2, 5, 3)$. In contrast, for the simplification of unweighted directed graphs, references [1, 2] consider the partitioning of V into a different type equivalence classes which are the strongly connected components of the graph. For this example graph, there is only one strongly connected component which is the set of all nodes. This is different from the two-part equivalence class partitioning induced by the \sim relation considered in this paper. The intra-equivalence class edge sets are $E_1 = \emptyset$, $E_2 = \{(2, 5), (3, 2), (3, 4), (4, 2), (5, 3)\}$ and $E_2^r = \{(3, 2)\}$. The inter-equivalence class edge sets are $E_{12} = \{(1, 2)\}$ and $E_{21} = \{(3, 1)\}$. The representing edges for E_{12} and E_{21} are, respectively, $(v_{12}^1, v_{12}^2) = (1, 2)$ and $(v_{21}^2, v_{21}^1) = (3, 1)$. In this example, $E_0 = E_0^c = E_0^d = \{(1, 2), (3, 1)\}$.

- According to the definition of v_{ij}^i and v_{ij}^j in Definition 7.d, it holds that

$$\tilde{c}_{v_i v_j} = d_{v_i v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v_j}. \quad (28)$$

- The vector of edge weights is denoted $\tilde{c}(\tilde{E}_0)$.

Remark 6. For a graph G , different designations of the representing nodes in the equivalence classes (e.g., v_1, v_2, \dots, v_K) can result in different condensations \tilde{G} . However, certain properties of \tilde{G} vital to the main results in this paper are independent of the designation. See Remarks 8 and 12 for details.

The condensation of the graph in Figure 3 is illustrated in Figure 4.

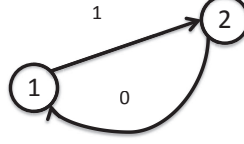


Figure 4: The condensation of the graph in Figure 3. There are two edges in the condensation: $(v_1, v_2) = (1, 2)$ and $(v_2, v_1) = (2, 1)$ (i.e., $\tilde{E}_0 = \{(1, 2), (2, 1)\}$). Their weights are $\tilde{c}_{v_1 v_2} = d_{v_1 v_{12}}^1 + c_{v_{12} v_{12}}^1 + d_{v_{12} v_2}^2 = 0 + 1 + 0 = 1$, and $\tilde{c}_{v_2 v_1} = d_{v_2 v_{21}}^2 + c_{v_{21} v_{21}}^1 + d_{v_{21} v_1}^1 = -2 + 2 + 0 = 0$.

The equivalence relation \sim (Definition 5), the equivalence classes (Definition 7) and the condensation (Definition 8) satisfy certain properties that will be useful in the proof of the main results in this subsection. Lemma 8 and Lemma 9 specify that relation \sim and the corresponding equivalence class partitioning are preserved even if a subset of edges is removed from the graph, as long as the removed edges form a redundant edge set.

Lemma 8. *Let $G = (V, E, c(E))$ denote an edge weighted directed graph without negative weight closed walks, and let R be any redundant edge set of G (Definition 3). Denote $G^c = (V, E \setminus R, c(E \setminus R))$. Let $i \in V$ and $j \in V$. Then $i \sim j$ in G if and only if $i \sim j$ in G^c .*

Proof. The proof considers only the case when $i \neq j$, since the statement is true by Definition 5 when $i = j$. If $i \sim j$ in G^c then i and j are on a zero-weight closed walk in G^c . The same closed walk is also in G because $(E \setminus R) \subseteq E$. Hence, $i \sim j$ in G . On the other hand, if $i \sim j$ in G , then there exists a zero-weight closed walk w in G traversing i and j . Part of w can be edges in R . However, for each $(u, v) \in R$ such that (u, v) is part of w , a replacement path p_{uv} exists in G^c such that $c_{p_{uv}} \leq c_{uv}$. By substituting edges in R which are part of w with the corresponding replacement paths in G^c , it is possible to construct another closed walk \hat{w} (in G^c) traversing i and j satisfying $c_{\hat{w}} \leq c_w = 0$. The no-negative-weight-cycle assumption in the statement excludes the case where $c_{\hat{w}} < 0$. Hence, \hat{w} is a zero-weight closed walk in G^c traversing i and j . In other words, $i \sim j$ in G^c . \square

Lemma 9. *Let $G = (V, E, c(E))$ an edge weighted directed graph without negative weight closed walks. The equivalence class partitioning defined by relation \sim is the same for all subgraphs $(V, E \setminus R, c(E \setminus R))$, where $R \subseteq E$ is any redundant edge set in G .*

Proof. This is a direct consequence of Lemma 8. \square

The following preliminary statements are also useful in the proof of the main results. Lemma 10 establishes some properties regarding the minimum walk weights (and the corresponding walks) between nodes in an equivalence class. Lemma 11 specifies that the cycles in the condensation are always positively weighted, even if the graph from which the condensation is derived can have zero-weight cycles.

Lemma 10. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. For $i, j \in V$, let d_{ij} be the minimum walk weight from i to j among all walks in G (i.e., Definition 6). Let $V' \subseteq V$, $E' \subseteq E$. Let $U \subseteq V'$ be an equivalence class in $(V', E', c(E'))$ defined by relation \sim . Define $E_U := \{(i, j) \in E' \mid i \in U, j \in U\}$, and $G_U = (U, E_U, c(E_U))$. For all $i \in U$, $j \in U$, the following statements hold:*

10.a *There exists a walk w_{ij} in G_U (which is a subgraph of G' which in turn is a subgraph of G) attaining the minimum weight d_{ij} among all walks in G which goes from i to j . Similiarly, there exists a walk w_{ji} in G_U attaining the minimum weight d_{ji} among all walks in G which goes from j to i . In case $i = j$, $d_{ii} = 0$ is attained by the degenerate walk containing the single node i .*

10.b $d_{ij} = -d_{ji}$.

10.c If $s \in U$, then $d_{ij} = d_{is} + d_{sj}$.

Proof. First, 10.a and 10.b are shown together. If $i = j$, then $d_{ij} = -d_{ji} = 0$. Thus, we only consider the case when $i \neq j$. By Definition 5, $i \in U$ and $j \in U$ means that there exists a zero-weight closed walk $w_{ii} : i \rightsquigarrow j \rightsquigarrow i$ in G_U . This closed walk can be decomposed into two walks in G_U : $w_{ij} : i \rightsquigarrow j$ and $w_{ji} : j \rightsquigarrow i$ (there can be multiple ways to decompose). Let $c_{w_{ii}} = 0$, $c_{w_{ij}}$ and $c_{w_{ji}}$ denote the weights of the walks respectively. Then the decomposition of w_{ii} implies that $c_{w_{ij}} + c_{w_{ji}} = c_{w_{ii}} = 0$. Hence, $c_{w_{ij}} = -c_{w_{ji}}$. Next, we show that indeed $c_{w_{ij}} = d_{ij}$ and $c_{w_{ji}} = d_{ji}$. First, note that $c_{w_{ij}} \geq d_{ij}$ by Definition 6. Suppose $c_{w_{ij}} > d_{ij}$, and hence there exists another walk \hat{w}_{ij} in G going from i and j such that $c_{\hat{w}_{ij}} < c_{w_{ij}}$. Then, concatenating \hat{w}_{ij} and w_{ji} leads to a closed walk with weight $c_{\hat{w}_{ij}} + c_{w_{ji}} = c_{\hat{w}_{ij}} - c_{w_{ij}} < 0$. This violates the no negative weight closed walk assumption in the statement. Thus, $c_{w_{ij}} = d_{ij}$. With a symmetric argument, it can be shown that $c_{w_{ji}} = d_{ji}$.

For 10.c, if $s = i$, $s = j$ or $i = j$ then the equality trivially holds because $d_{ii} = d_{jj} = 0$, and $d_{is} = -d_{si}$ by 10.b. Thus, we consider only the case where

s, i, j are all distinct. Since $i, j, s \in U$, by 10.a there exist walks $w_{is} : i \rightsquigarrow s$ and $w_{sj} : s \rightsquigarrow j$ with weights d_{is} and d_{sj} , respectively. Thus, concatenating w_{is} and w_{sj} yields a walk $i \rightsquigarrow s \rightsquigarrow j$ with weight $d_{is} + d_{sj}$. Consequently, by Definition 6, $d_{ij} \leq d_{is} + d_{sj}$. Next, we show that $d_{ij} < d_{is} + d_{sj}$ is impossible. Assume, on the contrary, that

$$d_{ij} < d_{is} + d_{sj} \quad (29)$$

holds. Let w_{ij} be a minimum weight walk in G attaining weight d_{ij} by Definition 6. In addition, since $i, j, s \in U$, by 10.a and 10.b there exist walks in G_U (and hence in G) $w_{js} : j \rightsquigarrow s$ and $w_{si} : s \rightsquigarrow i$ with weights $d_{js} = -d_{sj}$ and $d_{si} = -d_{is}$ respectively. Consequently, by concatenating w_{ij} , w_{js} and w_{si} we obtain a closed walk (in G) $i \rightsquigarrow j \rightsquigarrow s \rightsquigarrow i$ with weight $c_{w_{ij}} + c_{w_{js}} + c_{w_{si}} = d_{ij} - d_{sj} - d_{is} < 0$ according to (29). This violates the no negative weight closed walk assumption in the statement. Therefore, (29) does not hold, and $d_{ij} = d_{is} + d_{sj}$ as desired. \square

Lemma 11. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. Let $\tilde{G} := (\tilde{V}, \tilde{E}_0, \tilde{c}(\tilde{E}_0))$ be the condensation of G defined in Definition 8. Then, the weights of all cycles in \tilde{G} are positive.*

Proof. Let \tilde{w} denote a cycle in \tilde{G} as $(v_{i_0}, v_{i_1}, \dots, v_{i_m} = v_{i_0})$. By definition of cycle, $m \geq 2$ and at least one v_{i_q} for $q > 0$ is different from v_{i_0} . The weight of the cycle is

$$\tilde{c}_{\tilde{w}} = \tilde{c}_{v_{i_0} v_{i_1}} + \tilde{c}_{v_{i_1} v_{i_2}} + \dots + \tilde{c}_{v_{i_{m-1}} v_{i_m}}. \quad (30)$$

By (28), the edge weights are

$$\tilde{c}_{v_{i_k} v_{i_{k+1}}} = d_{v_{i_k} v_{i_k}^{i_k} v_{i_k}^{i_{k+1}}} + c_{v_{i_k}^{i_k} v_{i_k}^{i_{k+1}} v_{i_k}^{i_{k+1}}} + d_{v_{i_k}^{i_{k+1}} v_{i_k}^{i_{k+1}} v_{i_k}^{i_{k+1}}}, \quad \forall k \in \{0, \dots, m-1\}. \quad (31)$$

With (31), the expression in (30) can be rewritten as

$$\begin{aligned} \tilde{c}_{\tilde{w}} = & d_{v_{i_0} v_{i_0}^{i_0} v_{i_0}^{i_1}} + c_{v_{i_0}^{i_0} v_{i_0}^{i_1} v_{i_0}^{i_1}} + d_{v_{i_0}^{i_1} v_{i_0}^{i_1} v_{i_1}} + d_{v_{i_1} v_{i_1}^{i_1} v_{i_1}^{i_2}} + c_{v_{i_1}^{i_1} v_{i_1}^{i_2} v_{i_1}^{i_2}} + d_{v_{i_1}^{i_2} v_{i_1}^{i_2} v_{i_2}} \\ & + \dots + d_{v_{i_{m-1}} v_{i_{m-1}}^{i_{m-1}} v_{i_{m-1}}^{i_m}} + c_{v_{i_{m-1}}^{i_{m-1}} v_{i_{m-1}}^{i_m} v_{i_{m-1}}^{i_m}} + d_{v_{i_{m-1}}^{i_m} v_{i_{m-1}}^{i_m} v_{i_m}}. \end{aligned} \quad (32)$$

The right-hand side of (32) is the weight of a closed walk in G of the form

$$v_{i_0} \rightsquigarrow v_{i_0}^{i_0} \rightarrow v_{i_0}^{i_1} \rightsquigarrow v_{i_1} \rightsquigarrow v_{i_1}^{i_1} \rightarrow v_{i_1}^{i_2} \rightsquigarrow \dots \rightarrow v_{i_{m-1}}^{i_0} \rightsquigarrow v_{i_0}, \quad (33)$$

where the existence of the intra equivalence class walks and the inter equivalence class edges is guaranteed by Lemma 10.a and Definition 8, respectively.

$\tilde{c}_{\tilde{w}} < 0$ is impossible because of the statement assumption. In addition, if $\tilde{c}_{\tilde{w}} = 0$ then the closed walk (in G) in (33) would have zero weight. Consequently, v_{i_0}, v_{i_1}, \dots , which are the representing nodes of different equivalence classes, would be all traversed by one zero-weight closed walk. This is a contradiction. Therefore, $\tilde{c}_{\tilde{w}} > 0$ as desired. \square

Now we begin to analyze and characterize the maximum redundant edge set problem for an edge weighted directed graph $G = (V, E, c(E))$. The main result in this subsection is concerned with a decomposition of the set of decision variables (i.e., E), induced by the equivalence class partitioning in Definition 7:

$$E = E_0 \cup (E_1^r \cup E_2^r \cup \dots \cup E_K^r) \cup (E_1^c \cup E_2^c \cup \dots \cup E_K^c). \quad (34)$$

In the following, Lemma 12 and Lemma 13 are first introduced as components of the proof of subsequent lemmas. After that, Lemma 14 states that $(E_1^r \cup E_2^r \cup \dots \cup E_K^r)$ should always be included in any maximum redundant edge set of G . Lemma 15 delivers a similar but less straightforward result. Apart from other properties to be discussed, Lemma 15 states that at most one member for each E_{ij} (recall that $E_0 = \bigcup_{i,j} E_{ij}$) can be excluded from any maximum redundant edge set.

Lemma 12. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks, and let R be a redundant edge set of G . Suppose $A \subseteq E$ satisfies the property that for each $(i, j) \in A$ with edge weight c_{ij} , the following condition holds:*

$$\exists w_{ij} : i \rightsquigarrow j \text{ in } \left(V, E \setminus (R \cup A), c(E \setminus (R \cup A)) \right) \text{ such that } c_{w_{ij}} \leq c_{ij}. \quad (35)$$

Then, $R \cup A$ is also a redundant edge set of G .

Proof. If $R = \emptyset$ then the statement is true because (35) is a restatement of Definition 3 for A . Similarly, the statement holds trivially when $A = \emptyset$. Hence, for the rest of the proof we assume $R \neq \emptyset$, $A \neq \emptyset$. Since R is a redundant edge set of G , by Definition 3 for each $(u, v) \in R$ there exists a path $\hat{p}_{uv} : u \rightsquigarrow v$ in $(V, E \setminus R, c(E \setminus R))$ satisfying $c_{\hat{p}_{uv}} \leq c_{uv}$. The path \hat{p}_{uv} might include as parts the edges in A . However, we can substitute each $(i, j) \in A$ that is part of \hat{p}_{uv} with the corresponding replacement walk w_{ij} in (35). The outcome is a walk $w_{uv} : u \rightsquigarrow v$ in $(V, E \setminus (R \cup A), c(E \setminus (R \cup A)))$ such that $c_{w_{uv}} \leq c_{\hat{p}_{uv}} \leq c_{uv}$. Furthermore, by applying Lemma 3 with w_{uv} ,

we establish that for all $(u, v) \in R$,

$$\exists p_{uv} : u \rightsquigarrow v \text{ in } \left(V, E \setminus (R \cup A), c(E \setminus (R \cup A)) \right) \text{ s.t. } c_{p_{uv}} \leq c_{uv}. \quad (36)$$

Combing (35) (again, with an application of Lemma 3) and (36) yields the desired statement that $R \cup A$ is a redundant edge set of G . \square

Lemma 13. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. For $u, v \in V$, let d_{uv} be the minimum walk weight from u to v defined in Definition 6 for G . Let R be any redundant edge set of G . If $(i, j) \in E$ with edge weight satisfying $c_{ij} > d_{ij}$, then $R \cup \{(i, j)\}$ is a redundant edge set of G .*

Proof. $c_{ij} > d_{ij}$ implies that there exists a walk $w_{ij} : i \rightsquigarrow j$ in G with weight $c_{w_{ij}} < c_{ij}$. The walk w_{ij} might contain edges in R . However, by Definition 3 each edge of w_{ij} that is in R can be replaced by another path in $(V, E \setminus R, c(E \setminus R))$ with no greater weight. Hence, there exists a walk \hat{w}_{ij} in $(V, E \setminus R, c(E \setminus R))$ such that $c_{\hat{w}_{ij}} < c_{ij}$. If (i, j) is part of \hat{w}_{ij} , then \hat{w}_{ij} is of the form $i \rightsquigarrow i \rightarrow j \rightsquigarrow j$. The fact that $c_{\hat{w}_{ij}} < c_{ij}$ implies that at least one of the closed walks $i \rightsquigarrow i$ and $j \rightsquigarrow j$ must have negative weight. This contradicts the statement assumption. Hence, (i, j) cannot be part of \hat{w}_{ij} . Therefore, it holds that

$$\exists \hat{w}_{ij} : i \rightsquigarrow j \text{ in } \left(V, E \setminus (R \cup \{(i, j)\}), c(E \setminus (R \cup \{(i, j)\})) \right) \text{ s.t. } c_{\hat{w}_{ij}} \leq c_{ij}. \quad (37)$$

(37) implies that applying Lemma 12 with $A = \{(i, j)\}$ yields the desired statement. \square

Lemma 14. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. Let $(i, j) \in E_k^r$ for some $k \in \{1, 2, \dots, K\}$, where E_k^r is defined in (23) in Definition 7.c, in the context of G . Then for any redundant edge set (of G) denoted R , the union $R \cup \{(i, j)\}$ is also a redundant edge set of G . Consequently, $R \cup E_1^r \cup E_2^r \cup \dots \cup E_K^r$ is also a redundant edge set of G .*

Proof. This is a direct consequence of Lemma 13 because by (23) in Definition 7.c $c_{ij} > d_{ij}$ for $(i, j) \in E_k^r$, with d_{ij} being the minimum walk weight from i to j (among all walks in G) defined in Definition 6. \square

Lemma 15. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. For any $k \in \{1, 2, \dots, K\}$, $q \in \{1, 2, \dots, K\}$,*

$k \neq q$, let E_{kq} and E_{kq}^c be defined in Definition 7.d such that $E_{kq} \neq \emptyset$. If R is a redundant edge set of G , then for any $(i, j) \in E_{kq}^c$, the set $(R \cup E_{kq}) \setminus \{(i, j)\}$ is a redundant edge set of G .

Proof. Since $(R \cup E_{kq}) \setminus \{(i, j)\} = (R \setminus \{(i, j)\}) \cup (E_{kq} \setminus \{(i, j)\}) := R' \cup E'_{kq}$, we will show that $R' \cup E'_{kq}$ is a redundant edge set according to Definition 3. First, it is claimed that for each $(u, v) \in E'_{kq}$, there exists a replacement walk $w_{uv} : u \rightsquigarrow i \rightarrow j \rightsquigarrow v$ satisfying

$$w_{uv} \text{ in } (V, ((E_k \cup E_q) \setminus R') \cup \{(i, j)\}, c(((E_k \cup E_q) \setminus R') \cup \{(i, j)\})), \quad (38a)$$

$$\text{the weight of } w_{uv}, \text{ denoted } c_{w_{uv}}, \text{ satisfies } c_{w_{uv}} \leq c_{uv}. \quad (38b)$$

The argument for the existence of w_{uv} and (38a) is as follows: associated with E_{kq} let $[v_k]$ and $[v_q]$ be the equivalence classes in G , as defined in Definition 7.b. Since $R' \subseteq R$ and R is a redundant edge set of G , R' is also a redundant edge set of G . Hence, $[v_k]$ and $[v_q]$ remain equivalence classes in $(V, E \setminus R', c(E \setminus R'))$. Consequently, Lemma 10.a implies that there is a walk $u \rightsquigarrow i$ in $([v_k], E_k \setminus R', c(E_k \setminus R'))$. In addition, the weight of the walk is d_{ui} , the minimum walk weight $u \rightsquigarrow i$ in G in Definition 6. Similarly, Lemma 10.a implies that there is a walk $j \rightsquigarrow v$ in $([v_q], E_q \setminus R', c(E_q \setminus R'))$ with walk weight d_{jv} . Combining the walks $u \rightsquigarrow i$, $j \rightsquigarrow v$ and the edge (i, j) , we conclude that w_{uv} exists and (38a) is satisfied. To show (38b), first note that

$$c_{w_{uv}} = d_{ui} + c_{ij} + d_{jv}, \quad (39)$$

where c_{ij} is the weight of edge (i, j) . By Lemma 10.c, it holds that $d_{ui} = d_{uv_k} + d_{v_k i}$ since $u, i, v_k \in [v_k]$. Similarly, it holds that $d_{jv} = d_{jv_q} + d_{v_q v}$. Hence, (39) can be rewritten as

$$\begin{aligned} c_{w_{uv}} &= d_{uv_k} + \overbrace{d_{v_k i} + c_{ij} + d_{jv_q}}^{= \tilde{c}_{v_k v_q} \leq d_{v_k u} + c_{uv} + d_{v_q v} \text{ by (25) in Definition 7.d}} + d_{v_q v} \\ &\leq \underbrace{d_{uv_k} + d_{v_k u}}_{= 0 \text{ by Lemma 10.b}} + c_{uv} + \underbrace{d_{v_q v} + d_{v_q v}}_{= 0 \text{ by Lemma 10.b}} \\ &= c_{uv}. \end{aligned}$$

Therefore, a replacement walk $w_{uv} : u \rightsquigarrow i \rightarrow j \rightsquigarrow v$ satisfying (38a) and (38b) exists. Since it holds that

$$(i, j) \notin R' \implies R' \setminus (i, j) = R', \text{ and } E'_{kq} \cap (E_k \cup E_q \cup \{(i, j)\}) = \emptyset, \quad (40)$$

The set $((E_k \cup E_q) \setminus R') \cup \{(i, j)\}$ can be rewritten as

$$\begin{aligned}
& ((E_k \cup E_q) \setminus R') \cup \{(i, j)\} \\
&= (E_k \cup E_q \cup \{(i, j)\}) \setminus (R' \setminus \{(i, j)\}) \\
&\stackrel{(40)}{=} (E_k \cup E_q \cup \{(i, j)\}) \setminus (R' \cup E'_{kq}) \\
&\subseteq E \setminus (R' \cup E'_{kq}).
\end{aligned}$$

This implies that (38a) can be modified to state that

$$w_{uv} \text{ is in } (V, E \setminus (R' \cup E'_{kq}), c(E \setminus (R' \cup E'_{kq}))). \quad (41)$$

Finally, (41) and (38b) imply that (35) in Lemma 12 holds with $R = R'$ and $A = E'_{kq}$. Hence, Lemma 12 guarantees that $R' \cup E'_{kq}$ is a redundant edge set of G . \square

The implication of Lemma 14 and Lemma 15 is as follows: corresponding to the edge set decomposition in (34), any maximum redundant edge set must be a member of

$$\mathcal{R} = \left\{ R_0 \cup \left(\bigcup_{k=1}^K E_k^r \cup R_k \right) \right\} = \left\{ \left(\bigcup_{1 \leq i \neq j \leq K} R_{ij} \right) \cup \left(\bigcup_{k=1}^K E_k^r \cup R_k \right) \right\} \quad (42)$$

where

$$\begin{aligned}
& \text{For } k \geq 1, \quad E_k^r \text{ is defined by (23) in Definition 7.c,} \\
& R_0 \subseteq E_0, \quad R_0 = \bigcup_{1 \leq i \neq j \leq K} R_{ij}, \\
& R_{ij} \subseteq E_{ij}, \quad |E_{ij} \setminus R_{ij}| \leq 1, \quad (E_{ij} \setminus R_{ij}) \subseteq E_{ij}^c, \\
& R_k \subseteq E_k^c, \quad k \in \{1, 2, \dots, K\}.
\end{aligned} \quad (43)$$

In (42) the inclusion of E_k^r for $1 \leq k \leq K$ is due to Lemma 14. For now, R_{ij} and R_k are not fully specified, and the partial characterization of R_{ij} in (43) is due to Lemma 15. Furthermore, Lemma 15 suggests that, instead of searching over \mathcal{R} for a maximum redundant edge set, it is without loss of generality to search over the following restricted set

$$\begin{aligned}
\mathcal{R}^d &:= \left\{ \left(\bigcup_{1 \leq i \neq j \leq K} ((E_{ij} \setminus \{(v_{ij}^i, v_{ij}^j)\}) \cup R_{ij}^d) \right) \cup \left(\bigcup_{k=1}^K E_k^r \cup R_k \right) \right\} \\
&= \left\{ (E_0 \setminus E_0^d) \cup R_0^d \cup \left(\bigcup_{k=1}^K E_k^r \cup R_k \right) \right\},
\end{aligned} \quad (44)$$

where

$$R_{ij}^d \subseteq \{(v_{ij}^i, v_{ij}^j)\}, \quad R_0^d = \bigcup_{1 \leq i \neq j \leq K} R_{ij}^d, \quad R_0^d \subseteq E_0^d, \quad R_k \subseteq E_k^c,$$

and we note that E_0^d (i.e., the collection of all edges (v_{ij}^i, v_{ij}^j)) is defined in (26). The restriction from (42) to (44) amounts to the following specializations

$$\begin{aligned} R_{ij} \text{ in (42)} &\rightarrow (E_{ij} \setminus \{(v_{ij}^i, v_{ij}^j)\}) \cup R_{ij}^d \text{ in (44)}, \\ R_0 \text{ in (42)} &\rightarrow (E_0 \setminus E_0^d) \cup R_0^d \text{ in (44)}. \end{aligned}$$

The restriction is justified as follows: suppose $R^* \in \mathcal{R}$ in (42),

$$R^* = \left\{ \left(\bigcup_{1 \leq i \neq j \leq K} R_{ij}^* \right) \cup \left(\bigcup_{k=1}^K E_k^r \cup R_k^* \right) \right\}$$

is a maximum redundant edge set, with appropriate choices (to be discussed in the sequel) of R_{ij}^* and R_k^* satisfying (43). Define R_{ij}^{d*} by

$$R_{ij}^{d*} := \begin{cases} \{(v_{ij}^i, v_{ij}^j)\}, & \text{if } R_{ij}^* = E_{ij}^c, \\ \emptyset, & \text{if } R_{ij}^* \neq E_{ij}^c, \end{cases}$$

and define R^{d*} by

$$R^{d*} = \left\{ \left(\bigcup_{1 \leq i \neq j \leq K} ((E_{ij} \setminus \{(v_{ij}^i, v_{ij}^j)\}) \cup R_{ij}^{d*}) \right) \cup \left(\bigcup_{k=1}^K E_k^r \cup R_k^* \right) \right\}.$$

By construction, $R^{d*} \in \mathcal{R}^d$ in (44) and $|R^{d*}| = |R^*|$. In addition, since R^* is a maximum redundant edge set, Lemma 15 states that R^{d*} is also a redundant edge set (and $|R^{d*}| = |R^*|$). Hence, R^{d*} is a maximum redundant edge set. Conversely, define the set-valued function $F : \mathcal{R}^d \mapsto 2^{\mathcal{R}}$,

$$F(X) = \left\{ R \subseteq \mathcal{R} \left| \begin{array}{l} R = \left(\bigcup_{1 \leq i \neq j \leq K} R_{ij} \right) \cup \left(\bigcup_{k=1}^K (E_k^r \cup R_k) \right), \\ X = \left(\bigcup_{1 \leq i \neq j \leq K} ((E_{ij} \setminus \{(v_{ij}^i, v_{ij}^j)\}) \cup R_{ij}^d) \right) \cup \left(\bigcup_{k=1}^K (E_k^r \cup R_k) \right), \\ R_{ij} := \begin{cases} E_{ij}, & \text{if } R_{ij}^d \neq \emptyset \\ E_{ij} \setminus \{(g, h)\}, & (g, h) \in E_{ij}^c, \text{ if } R_{ij}^d = \emptyset \end{cases} \end{array} \right. \right\}. \quad (45)$$

Then, if $R^{d\star} \in \mathcal{R}^d$ in (44) is a maximum redundant edge set, by Lemma 15 the set $F(R^{d\star}) \subseteq \mathcal{R}$ is the set of all maximum redundant edge sets of the form in (42) that share the same R_k^\star 's as in $R^{d\star}$.

Next, we focus on the specialized maximum redundant edge sets in \mathcal{R}^d in (44). Let R^d denote any member of \mathcal{R}^d . The second expression in (44) reveals that the components of R^d which are not fully specified are $R_0^d \subseteq E_0^d$ and $R_k \subseteq E_k^c$ for $k = 1, 2, \dots, K$. First, we examine the conditions on these components under which R^d is a redundant edge set (of graph $G = (V, E, c(E))$) according to Definition 3. Since the fixed components in (44) (i.e., the edges guaranteed to be included in R^d) can be written as

$$\begin{aligned} \left(\bigcup_{k=1}^K E_k^r \right) \cup (E_0 \setminus E_0^d) &= \left(\bigcup_{k=1}^K (E_k \setminus E_k^c) \right) \cup (E_0 \setminus E_0^d) \\ &= E \setminus (E_0^d \cup \left(\bigcup_{k=1}^K E_k^c \right)) \\ &:= E \setminus \mathcal{E}, \end{aligned} \quad (46)$$

two necessary conditions for R^d to be a redundant edge set are

$$\begin{aligned} \forall (u, v) \in R_0^d, \exists p_{uv} : u \rightsquigarrow v \text{ in} \\ \left(V, \mathcal{E} \setminus (R_0^d \cup \left(\bigcup_{k=1}^K R_k \right)), c \left(\mathcal{E} \setminus (R_0^d \cup \left(\bigcup_{k=1}^K R_k \right)) \right) \right), \text{ with } c_{p_{uv}} \leq c_{uv}, \end{aligned} \quad (47)$$

and

$$\begin{aligned} \forall k \in \{1, 2, \dots, K\}, \forall (u, v) \in R_k, \exists p_{uv} : u \rightsquigarrow v \text{ in} \\ \left(V, \mathcal{E} \setminus (R_0^d \cup \left(\bigcup_{k=1}^K R_k \right)), c \left(\mathcal{E} \setminus (R_0^d \cup \left(\bigcup_{k=1}^K R_k \right)) \right) \right), \text{ with } c_{p_{uv}} \leq c_{uv}. \end{aligned} \quad (48)$$

Conversely, if (47) and (48) are satisfied then Lemma 12 can be applied to show that R^d is indeed a redundant edge set of G . Lemma 12 is applied in the following settings:

$$R \leftarrow \left(\bigcup_{k=1}^K E_k^r \right) \cup (E_0 \setminus E_0^d) = E \setminus \mathcal{E}, \quad A \leftarrow R_0^d \cup \left(\bigcup_{k=1}^K R_k \right), \quad (35) \leftarrow (47), (48),$$

and the fact that $(\bigcup_{k=1}^K E_k^r) \cup (E_0 \setminus E_0^d)$ is a redundant edge set of G is due to Lemma 14 and Lemma 15. Therefore, R^d is a redundant edge set of G if and only if (47) and (48) are satisfied. The following two statements, Lemma 16 and Lemma 17, specify that the conditions in (47) and (48) are in fact equivalent to $K + 1$ decoupled conditions, one for each set of $R_0^d, R_1, R_2, \dots, R_K$. These two statements are first described. Then, their consequences are discussed.

Lemma 16. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks, and let $(i, j) \in E_k^c$ for some $k \in \{1, 2, \dots, K\}$ (see (24) in Definition 7.c for E_k^c). Let the minimum walk weight d_{ij} (of G) be defined in Definition 6. If $w_{ij} : i \rightsquigarrow j$ is a walk in G such that $c_{w_{ij}} = d_{ij}$ (note that $c_{w_{ij}} \geq d_{ij}$ must hold), then all nodes traversed by w_{ij} are in the equivalence class $[v_k]$ (corresponding to E_k^c).*

Proof. By definition $(i, j) \in E_k^c$ means that $i, j \in [v_k]$. Hence, Lemma 10.a and 10.b states that there exists a walk $w_{ji} : j \rightsquigarrow i$ (in G) such that $c_{w_{ji}} = -d_{ij} = -c_{ij}$ (the last equality is due to the fact that $(i, j) \in E_k^c$). Let w_{ij} be the walk described in the statement (with weight $c_{w_{ij}} = d_{ij} = c_{ij}$). If w_{ij} traverses a node $t \notin [v_k]$, then by concatenating w_{ij} and w_{ji} we obtain a closed walk $w_{tt} : i \rightsquigarrow t \rightsquigarrow j \rightsquigarrow i$. The weight of w_{tt} is $c_{w_{tt}} = c_{w_{ij}} + c_{w_{ji}} = c_{ij} - c_{ij} = 0$. Therefore, the assumption that $t \notin [v_k]$ leads to the contradictory conclusion that $t \sim i$ and hence $t \in [v_k]$. Consequently, the walk w_{ij} cannot traverse any node $t \notin [v_k]$. \square

Lemma 17. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. In addition, let the following be assumed in the context of G :*

1. $[v_1], [v_2], \dots, [v_K]$ denote the equivalence classes in V induced by relation \sim (see Definition 7.b).
2. E_0^d, E_k^c for $k = 1, 2, \dots, K$ are defined in Definition 7.e and 7.c, respectively. Let $\mathcal{E} := E_0^d \cup (\cup_{q=1}^K E_q^c)$.
3. For $k \in \{1, \dots, K\}$, let $R_k \subseteq E_k^c$ be given and assume that $[v_1], \dots, [v_K]$ remain equivalence classes in $(V, \mathcal{E} \setminus (\cup_{q=1}^K R_q), c(\mathcal{E} \setminus (\cup_{q=1}^K R_q)))$.
4. $\tilde{G} = (\tilde{V}, \tilde{E}_0, \tilde{c}(\tilde{E}_0))$ is the condensation of G , in accordance with the designation of representing nodes in $[v_1], \dots, [v_K]$. \tilde{E}_0 is the set of all edges in \tilde{G} (see Definition 8).
5. Let $R_0^d \subseteq E_0^d$ be given, and let $\tilde{R}_0 \subseteq \tilde{E}_0$ be defined to correspond to R_0^d in the sense that $(v_i, v_j) \in \tilde{R}_0$ if and only if $(v_{ij}^i, v_{ij}^j) \in R_0^d$.

Then, the following two statements are equivalent:

- (a) *For each $(v_{ij}^i, v_{ij}^j) \in R_0^d$, there is a (replacement) walk $w_{v_{ij}^i v_{ij}^j}$ in graph $(V, \mathcal{E} \setminus (R_0^d \cup (\cup_{k=1}^K R_k)), c(\mathcal{E} \setminus (R_0^d \cup (\cup_{k=1}^K R_k))))$ satisfying $c_{v_{ij}^i v_{ij}^j} \geq c_{w_{v_{ij}^i v_{ij}^j}}$.*

- (b) For each $(v_i, v_j) \in \tilde{R}_0$, there is a (replacement) walk $\tilde{w}_{v_i v_j}$ in graph $(\tilde{V}, \tilde{E}_0 \setminus \tilde{R}_0, \tilde{c}(\tilde{E}_0 \setminus \tilde{R}_0))$ satisfying $\tilde{c}_{v_i v_j} \geq \tilde{c}_{\tilde{w}_{v_i v_j}}$.

Proof. For convenience, we denote

$$\begin{aligned}\mathcal{G}^c &:= (V, \mathcal{E} \setminus (R_0^d \cup (\cup_{k=1}^K R_k)), c(\mathcal{E} \setminus (R_0^d \cup (\cup_{k=1}^K R_k)))), \\ \tilde{\mathcal{G}}^c &:= (\tilde{V}, \tilde{E}_0 \setminus \tilde{E}_r^0, \tilde{c}(\tilde{E}_0 \setminus \tilde{E}_r^0)).\end{aligned}$$

Because of the definition of E_0^d , in \mathcal{G}^c every walk $w_{v_{ij}^i v_{ij}^j}$ from v_{ij}^i to v_{ij}^j is of the form

$$\underbrace{v_{ij}^i \rightsquigarrow v_{k(1)k(2)}^{k(1)}}_{\text{in } \mathcal{G}_{k(1)}^c} \rightarrow \underbrace{v_{k(1)k(2)}^{k(2)} \rightsquigarrow v_{k(2)k(3)}^{k(2)}}_{\text{in } \mathcal{G}_{k(2)}^c} \rightarrow \cdots \rightarrow \underbrace{v_{k(m-1)k(m)}^{k(m)} \rightsquigarrow v_{ij}^j}_{\text{in } \mathcal{G}_{k(m)}^c}, \quad (49)$$

where $k(1) = i$, $k(m) = j$, $k(2), \dots, k(m-1) \in \{1, 2, \dots, K\}$ are indices of the intermediate equivalence classes in the order traversed by $w_{v_{ij}^i v_{ij}^j}$. In addition, $\mathcal{G}_{k(q)}^c = ([v_{k(q)}], E_{k(q)}^c \setminus R_{k(q)}, c(E_{k(q)}^c \setminus R_{k(q)}))$ for $q = 1, 2, \dots, m$. Due to statement assumption 3, $[v_{k(q)}]$'s remain equivalence classes in \mathcal{G}^c . Thus, by Lemma 10.a for any two nodes s and t in $[v_{k(q)}]$ there exists at least one walk w_{st} from s to t in $\mathcal{G}_{k(q)}^c$. Since $R_k \subseteq E_k^c$ and $E_{k(q)}^c \cap E_k^c = \emptyset$ as long as $k(q) \neq k$, it holds that $R_k \cap E_{k(q)}^c = \emptyset$ as long as $k(q) \neq k$. In addition, since $R_0^d \subseteq E_0^d$ and $E_0^d \cap E_k^c = \emptyset$ for $k \geq 1$, $R_0^d \cap E_k^c = \emptyset$ for $k \geq 1$. Therefore, it holds that

$$\underbrace{(E_{k(q)}^c \setminus R_{k(q)})}_{\text{edge set of } \mathcal{G}_{k(q)}^c} = (E_{k(q)}^c \setminus (R_0^d \cup (\cup_{k=1}^K R_k))) \subseteq \underbrace{(\mathcal{E} \setminus (R_0^d \cup (\cup_{k=1}^K R_k)))}_{\text{edge set of } \mathcal{G}^c}.$$

Thus, since w_{st} is in $\mathcal{G}_{k(q)}^c$ it is also in \mathcal{G}^c . Therefore, a walk of the form in (49) exists in \mathcal{G}^c if and only if all edges $(v_{k(q)k(q+1)}^{k(q)}, v_{k(q)k(q+1)}^{k(q+1)})$ exist in $E_0^d \setminus R_0^d$ for $q = 1, 2, \dots, m-1$, since these edges can only be contained in E_0^d or R_0^d . By Definition 8 and statement assumption 5, these edges exist if and only if the edges $(v_{k(q)}, v_{k(q+1)})$ exist in $\tilde{E}_0 \setminus \tilde{R}_0$ for all q . Further, if a walk $w_{v_{ij}^i v_{ij}^j}$ of the form (49) exists in \mathcal{G}^c then the corresponding walk $\tilde{w}_{v_i v_j}$ in $\tilde{\mathcal{G}}^c$ is of the form

$$(v_i =) v_{k(1)} \rightarrow v_{k(2)} \rightarrow \cdots \rightarrow v_{k(m)} (= v_j). \quad (50)$$

Conversely, if the walk $\tilde{w}_{v_i v_j}$ in (50) exists in $\tilde{\mathcal{G}}^c$, then in \mathcal{G}^c at least one walk $w_{v_{ij}^i v_{ij}^j}$ of the form (49) exists (the possible multiplicity of the walks is due to the possibilities of multiple walks within $\mathcal{G}_{k(q)}^c$).

Next, we establish the equivalence between the walk weight inequalities (i.e., $c_{v_{ij}^i v_{ij}^j} \geq c_{w_{v_{ij}^i v_{ij}^j}}$ and $\tilde{c}_{v_i v_j} \geq \tilde{c}_{\tilde{w}_{v_i v_j}}$). We consider only the cases when $w_{v_{ij}^i v_{ij}^j}$ is restricted to the choices where the walks in $\mathcal{G}_{k(q)}^c$ have minimum weights. By Lemma 10.a, these minimum weights are $d_{v_{ij}^i v_{k(1)k(2)}^{k(1)}}$, $d_{v_{k(m-1)k(m)}^{k(m)}}$, and $d_{v_{k(q-1)k(q)}^{k(q)} v_{k(q)k(q+1)}^{k(q)}}$ for $q = 2, 3, \dots, m-1$ respectively. Therefore, $c_{v_{ij}^i v_{ij}^j} \geq c_{w_{v_{ij}^i v_{ij}^j}}$ if and only if

$$\begin{aligned} c_{v_{ij}^i v_{ij}^j} &\geq d_{v_{ij}^i v_{k(1)k(2)}^{k(1)}} + d_{v_{k(m-1)k(m)}^{k(m)}} + \\ &+ \sum_{q=2}^{m-1} d_{v_{k(q-1)k(q)}^{k(q)} v_{k(q)k(q+1)}^{k(q)}} + \sum_{q=1}^{m-1} c_{v_{k(q)k(q+1)}^{k(q)} v_{k(q)k(q+1)}^{k(q+1)}}. \end{aligned} \quad (51)$$

By Lemma 10.b and 10.c, in (51) it holds that

$$\begin{aligned} d_{v_{ij}^i v_{k(1)k(2)}^{k(1)}} &= d_{v_{ij}^i v_{k(1)}} + d_{v_{k(1)} v_{k(1)k(2)}^{k(1)}} = -d_{v_{k(1)} v_{ij}^i} + d_{v_{k(1)} v_{k(1)k(2)}^{k(1)}}, \\ d_{v_{k(m-1)k(m)}^{k(m)}} &= d_{v_{k(m-1)k(m)}^{k(m)}} + d_{v_{k(m)} v_{ij}^j} = d_{v_{k(m-1)k(m)}^{k(m)}} - d_{v_{ij}^j v_{k(m)}}, \\ d_{v_{k(q-1)k(q)}^{k(q)} v_{k(q)k(q+1)}^{k(q)}} &= d_{v_{k(q-1)k(q)}^{k(q)}} + d_{v_{k(q)} v_{k(q)k(q+1)}^{k(q)}}, \quad \forall q = 2, \dots, m-1. \end{aligned} \quad (52)$$

Therefore, with (52) the inequality in (51) can be rearranged into

$$\begin{aligned} d_{v_{k(1)} v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v_{k(m)}} &\geq \sum_{q=1}^{m-1} \left(d_{v_{k(q)} v_{k(q)k(q+1)}^{k(q)}} + \right. \\ &\quad \left. c_{v_{k(q)k(q+1)}^{k(q)} v_{k(q)k(q+1)}^{k(q+1)}} + d_{v_{k(q)k(q+1)}^{k(q+1)} v_{k(q+1)}} \right). \end{aligned} \quad (53)$$

Since $i = k(1)$ and $j = k(m)$, by (28) in Definition 8, (53) can be rewritten as

$$\tilde{c}_{v_i v_j} = d_{v_{k(1)} v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v_{k(m)}} \geq \sum_{q=1}^{m-1} \tilde{c}_{v_{k(q)} v_{k(q+1)}} = \tilde{c}_{\tilde{w}_{v_i v_j}}, \quad (54)$$

where the last equality in (54) is due to (50). Therefore,

$$c_{v_{ij}^i v_{ij}^j} \geq c_{w_{v_{ij}^i v_{ij}^j}} \iff \tilde{c}_{v_i v_j} \geq \tilde{c}_{\tilde{w}_{v_i v_j}}.$$

Now we establish the equivalence between (a) and (b) in the statement. If R_0^d satisfies (a), then for each $(v_{ij}^i, v_{ij}^j) \in R_0^d$ there exists a walk $w_{v_{ij}^i v_{ij}^j}$ of

the form in (49) satisfying $c_{v_{ij}^i v_{ij}^j} \geq c_{w_{v_{ij}^i v_{ij}^j}}$. Additionally, we can assume in $w_{v_{ij}^i v_{ij}^j}$ all walks in $\mathcal{G}_{k(q)}^c$ for all q have the least possible weights. Therefore, by the previous parts of the proof, the corresponding edge (v_i, v_j) and walk $\tilde{w}_{v_i v_j}$ exists in \tilde{G}^c , and they satisfy the inequality $\tilde{c}_{v_i v_j} \geq \tilde{c}_{\tilde{w}_{v_i v_j}}$. Thus, $\tilde{R}_0 \subseteq \tilde{E}_0$, as defined in the statement, satisfies (b). This shows that (a) implies (b). The argument for (b) implying (a) can be shown in a similar fashion. \square

Now we analyze the consequences of Lemma 16 and Lemma 17. Due to Lemma 16, in condition (48) \mathcal{E} can be replaced with E_k^c . That is, (48) becomes

$$\begin{aligned} \forall k \in \{1, 2, \dots, K\}, \quad \forall (u, v) \in R_k, \quad \exists p_{uv} : u \rightsquigarrow v \\ \text{in } ([v_k], E_k^c \setminus (R_0^d \cup (\bigcup_{k=1}^K R_k)), c(E_k^c \setminus (R_0^d \cup (\bigcup_{k=1}^K R_k)))), \quad c_{p_{uv}} \leq c_{uv}. \end{aligned} \quad (55)$$

In addition, since $E_k^c \cap R_q = \emptyset$ as long as $q \neq k$ and $E_k^c \cap R_0^d = \emptyset$, it holds that $E_k^c \setminus (R_0^d \cup (\bigcup_{k=1}^K R_q)) = E_k^c \setminus R_k$. Therefore, (55) can be further simplified to establish the following observations:

$$R_k \text{ is part of redundant edge set } R^d, \text{ for } 1 \leq k \leq K \quad (56a)$$

$$\iff \forall (u, v) \in R_k, \exists p_{uv} \text{ in } ([v_k], E_k^c \setminus R_k, c(E_k^c \setminus R_k)), \quad c_{p_{uv}} \leq c_{uv} \quad (56b)$$

$$\iff R_k \text{ is a redundant edge set in } ([v_k], E_k^c, c(E_k^c)). \quad (56c)$$

One of the consequences of (56) is that, for $k = 1, 2, \dots, K$, whether or not R_k is part of a redundant edge set R^d does not depend on the choices of R_q for $q \in \{1, \dots, K\} \setminus \{k\}$ or the choice of R_0^d . In addition, (56) can be used to establish a similar independence result for R_0^d . To begin, notice that by Lemma 14 and Lemma 15, $(\bigcup_{k=1}^K E_k^r) \cup (E_0 \setminus E_0^d)$ is a redundant edge set in G . In addition, by (46) $\mathcal{E} := E_0^d \cup (\bigcup_{k=1}^K E_k^c) = E \setminus ((\bigcup_{k=1}^K E_k^r) \cup (E_0 \setminus E_0^d))$. Hence, Lemma 9 states that

$$[v_1], [v_2], \dots, [v_K] \text{ are equivalence classes in the graph } (V, \mathcal{E}, c(\mathcal{E})). \quad (57)$$

Secondly, in order for R^d to be a redundant edge set (in G), (56) must hold. Then, by (56b), $\bigcup_{k=1}^K R_k$ is a redundant edge set in $(V, \mathcal{E}, c(\mathcal{E}))$. Consequently, when applied to $(V, \mathcal{E}, c(\mathcal{E}))$, which is a subgraph of G without negative weight closed walks, Lemma 9 implies that

$$[v_1], \dots, [v_K] \text{ are equivalence classes in } (V, \mathcal{E} \setminus (\bigcup_{k=1}^K R_k), c(\mathcal{E} \setminus (\bigcup_{k=1}^K R_k))). \quad (58)$$

(58) implies that statement assumption 3 for Lemma 17 is satisfied with the \mathcal{E} and R_k for $k = 1, 2, \dots, K$. Therefore, with \tilde{R}_0 defined in statement assumption 5 in Lemma 17, the lemma specifies that

$$\begin{aligned} & R_0^d \text{ is part of a redundant edge set } R^d \\ \iff & \text{condition (47)} \\ \iff & \tilde{R}_0 \text{ is a redundant edge set in } \tilde{G}, \text{ the condensation of } G. \end{aligned} \quad (59)$$

According to Definition 8, \tilde{G} is independent of R_q for $q \in \{1, 2, \dots, K\}$. Hence, (59) establishes the desired property that whether or not R_0^d is part of a redundant edge set is independent of the choices of R_k for $k \in \{1, 2, \dots, K\}$. In conclusion, for $R^d = ((E_0 \setminus E_0^d) \cup R_0^d) \cup (\bigcup_{k=1}^K E_k^r \cup R_k)$ (i.e., (44)) to be a redundant edge set of G , it is necessary and sufficient for R_0^d to satisfy (59) and each of R_k (for $k = 1, 2, \dots, K$) to satisfy its individual version of (56).

The decoupling of the redundant edge set membership requirements in (56) and (59) suggests that the maximum redundant edge set problem can be decoupled into $K + 1$ independent maximum redundant edge set subproblems on the graphs $([v_k], E_k^c, c(E_k^c))$ in (56) and on the condensation \tilde{G} in (59), respectively. The following statement, whose proof has already been discussed, summarizes the main decomposition results which have been discussed so far:

Theorem 3. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. Let the following be defined in the context of G :*

- *K is the number of equivalence classes induced by relation \sim in Definition 7.a.*
- *For $k \in \{1, 2, \dots, K\}$, $[v_k]$ denotes the equivalence class defined in Definition 7.b.*
- *For $k \in \{1, 2, \dots, K\}$, E_k^r and E_k^c are defined in Definition 7.c.*
- *For $1 \leq i \neq j \leq K$, E_{ij} and E_{ij}^c are defined in Definition 7.d.*
- *\tilde{G} is the condensation of G defined in Definition 8.*

Then, every maximum redundant edge set of G can be parameterized by

$$R_0^* \cup \left(\bigcup_{k=1}^K (E_k^r \cup R_k^*) \right), \quad (60)$$

where for $k \in \{1, 2, \dots, K\}$, R_k^* is a maximum redundant edge set of the subgraph $([v_k], E_k^c, c(E_k^c))$. In addition, R_0^* is parameterized by

$$R_0^* = \bigcup_{1 \leq i \neq j \leq K} R_{ij}^*, \quad (61)$$

where

$$R_{ij}^* = \begin{cases} E_{ij}, & \text{if } (v_i, v_j) \in \tilde{R}_0^* \\ E_{ij} \setminus \{(g, h)\} & \text{for some } (g, h) \in E_{ij}^c, \text{ if } (v_i, v_j) \notin \tilde{R}_0^* \end{cases}, \quad (62)$$

and \tilde{R}_0^* is the maximum redundant edge set of \tilde{G} , the condensation of G .

Remark 7. The expression in (62) is jointly due to Lemma 17, (44) and (45).

Remark 8. While it appears that the statement of Theorem 3 (e.g., E_{ij}^c) depends on the choices of the representing nodes v_1, \dots, v_K in the equivalence classes, Theorem 3 in fact holds irrespective of these choices. In particular, it can be shown (in Appendix B) that

- The definition of E_{ij}^c is independent of the choices of v_1, \dots, v_K .
- With different choices of v_1, \dots, v_K , it is possible to define different condensations of G with different representing nodes and different edge weights \tilde{c} . However, for all $1 \leq i \neq j \leq K$, $|\tilde{R}_0^* \cap ([v_i] \times [v_j])|$ (which can only be 0 or 1) is independent of the choices of v_1, \dots, v_K .

As a result of Theorem 3, the graph in Figure 3 with its maximum redundant edge set removed is illustrated in Figure 5.

3.3 Computation for maximum redundant edge set

To compute the quantities in the statement of Theorem 3, the first step is the identification of the equivalence classes $[v_1], [v_2], \dots, [v_k]$ defined by relation \sim . The following statement is useful in the identification:

Lemma 18. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. For $i, j \in V$, let d_{ij} be the minimum walk weight defined in Definition 6. Then, $i \sim j$ if and only if $d_{ij} + d_{ji} = 0$.*

Proof. If $i = j$, then by Definition 5 $i \sim j$ and by Definition 6 $d_{ij} + d_{ji} = d_{ii} + d_{ii} = 0$. Hence the statement holds trivially when $i = j$. Next, we consider the case when $i \neq j$. If $i \sim j$, then Lemma 10.b specifies that

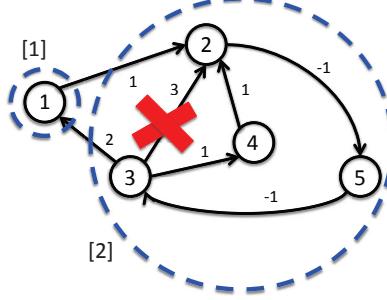


Figure 5: As a result of Theorem 3, the maximum redundant edge set of the graph in Figure 3 is $R^* = \{(3, 2)\}$ (in general the maximum redundant edge set need not be unique). In the parameterization in (60), $K = 2$, $E_1^r = \emptyset$ and $E_r^2 = \{(3, 2)\}$. $R_1^* = \emptyset$, and by inspection $R_2^* = \emptyset$ since the subgraph $([2], E_2^c = \{(2, 5), (5, 3), (3, 4), (4, 2)\}, c(E_2^c))$ is a zero-weight cycle. From Figure 4, the maximum redundant edge set of the condensation \tilde{G} is $\tilde{R}_0^* = \emptyset$. Hence, from (62) $R_{12}^* = R_{21}^* = \emptyset$ (since, for instance, $E_{12} = E_{12}^c = \{(1, 2)\}$).

$d_{ij} + d_{ji} = d_{ij} - d_{ij} = 0$. Conversely, suppose $d_{ij} + d_{ji} = 0$. By Definition 6, associated with d_{ij} and d_{ji} there exist walks $w_{ij} : i \rightsquigarrow j$ and $w_{ji} : j \rightsquigarrow i$ with weights d_{ij} and d_{ji} respectively. Concatenating w_{ij} and w_{ji} results in a zero weight closed walk traversing i and j , and hence $i \sim j$. \square

Based on Lemma 18, we identify the equivalence classes as follows:

Algorithm 2 (Identification of equivalence classes of graph $G = (V, E, c(E))$ without negative weight closed walks).

1. Solve the all-pair shortest path problem for all source/destination pairs in G . Let d_{ij} denote the shortest path distance from i to j .
2. For each pair of $1 \leq i \neq j \leq n$, declare $i \sim j$ if and only if $d_{ij} + d_{ji} = 0$. Build an undirected graph (V, E_\sim) such that edge $\{i, j\} \in E_\sim$ if and only if $i \sim j$ and $i \neq j$.
3. The equivalence classes defined by relation \sim are the connected components of (V, E_\sim) .

The first step of Algorithm 2 can be computed using Floyd-Warshall algorithm in $O(|V|^3)$ time, because G does not have any negative weight

closed walks. The second step requires $O(|V|^2)$ time. The third step requires $O(|V| + |E_{\sim}|) = O(|V|^2)$ time (e.g., [20]). Hence, Algorithm 2 requires $O(|V|^3)$ time. Once the equivalence classes have been identified, the computation involved in Definition 7 and Definition 8 requires $O(|E|)$ time and $O(K^2)$ time respectively.

Theorem 3 decomposes the maximum redundant edge set problem into $K + 1$ independent subproblems. Because of Lemma 11 and Theorem 1, solving the maximum redundant edge set subproblem on the condensation \tilde{G} (for R_0^*) requires only polynomial-time (i.e., $O(K^3)$ with $K \leq |V|$). On the other hand, to solve for R_k^* for $k = 1, 2, \dots, K$ in subgraphs $([v_k], E_k^c, c(E_k^c))$ is NP-hard. The argument is similar to the proof of Theorem 2: the minimum equivalent graph problem [1], even for strongly connected graphs, is NP-hard. Additionally, the maximum redundant edge set problem for graphs consisting only of one equivalence class generalizes the former problem. Hence, it is NP-hard to compute R_k^* . On the other hand, it turns out that the subproblem for finding a maximum redundant edge set in $([v_k], E_k^c, c(E_k^c))$ can be solved as the (NP-hard) minimum equivalent graph problem for undirected graph $([v_k], E_k^c)$ using available (exact or inexact) algorithms (e.g., [1, 22, 23]). The following statement provides the rationale:

Lemma 19. *Let $G = (V, E, c(E))$ be an edge weighted directed graph without negative weight closed walks. For $k \in \{1, 2, \dots, K\}$, let $[v_k]$ be the equivalence class defined by relation \sim (Definition 7.b), and E_k^c be defined in Definition 7.c. Let $R_k \subseteq E_k^c$ be given. Then, the following two statements are equivalent:*

- 19.a R_k is a redundant edge set of $([v_k], E_k^c, c(E_k^c))$.
- 19.b $([v_k], E_k^c)$ and $([v_k], E_k^c \setminus R_k)$ have the same reachability (i.e., there is a walk $i \rightsquigarrow j$ in $([v_k], E_k^c)$ if and only if there is a walk $i \rightsquigarrow j$ in $([v_k], E_k^c \setminus R_k)$).

Proof. We consider the case for $R_k \neq \emptyset$, since otherwise the statement is trivial. As argued in the proof of Theorem 2, condition 19.b is equivalent to

$$\forall (i, j) \in R_k, \text{ there exists a walk from } i \text{ to } j \text{ in } ([v_k], E_k^c \setminus R_k). \quad (63)$$

Hence, it suffices to argue for the equivalence between 19.a and (63). By Definition 3, condition 19.a implies (63). Conversely, if (63) holds then for each $(i, j) \in R_k$ there exists a walk w_{ij} in $([v_k], E_k^c \setminus R_k, c(E_k^c \setminus R_k))$. Let the walk w_{ij} be of the form $(i = i_0, i_1, \dots, i_m = j)$. By (24) and Lemma 10.c, the weight of walk w_{ij} is $c_{w_{ij}} = d_{i_0 i_1} + \dots + d_{i_{m-1} i_m} = d_{i_0 i_m} = d_{ij}$. This

is the same as the weight of edge $(i, j) \in E_k^c$ (again, by (24)). Hence, (63) implies 19.a, and the desired equivalence is established. \square

Remark 9. Lemma 19 states that for subgraph $([v_k], E_k^c, c(E_k^c))$, the maximum redundant edge set problem can be reduced to the minimum equivalent graph problem by ignoring the edge weights. However, this reduction is not guaranteed to be valid in more general cases. See Figure 6 for an example.

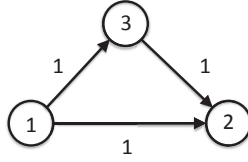


Figure 6: The graph illustrated is not one of $([v_k], E_k^c, c(E_k^c))$ for any k , since there is no zero weight cycle. When interpreted as a weighted graph, no edge is redundant according to Definition 3. However, when the edge weights are ignored, the minimum equivalent graph contains two edges $(1, 3)$ and $(3, 2)$. In other words, edge $(1, 2)$ is redundant in unweighted sense but not redundant in weighted sense.

The decomposition result (mainly Lemma 17) also leads to some guideline in obtaining approximate solutions to the maximum redundant edge set problem. If \hat{R}_k for $k = 1, 2, \dots, K$ are (not necessarily maximum) redundant edge sets for the subgraphs $([v_k], E_k^c, c(E_k^c))$, then it can be established that

$$[v_1], \dots, [v_K] \text{ are equivalence classes in } (V, \mathcal{E} \setminus (\bigcup_{k=1}^K \hat{R}_k), c(\mathcal{E} \setminus (\bigcup_{k=1}^K \hat{R}_k))),$$

where by (46) $\mathcal{E} = E_0^d \cup (\bigcup_{k=1}^K E_k^c)$. Hence, Lemma 17 can be applied to establish that the choices of redundant edge sets in E_0^d are independent of \hat{R}_k . Consequently, the largest cardinality redundant edge sets, given the components $\hat{R}_k \in E_k^c$ for $k = 1, 2, \dots, K$, can be parameterized as

$$R_0^* \cup \left(\bigcup_{k=1}^K (E_k^r \cup \hat{R}_k) \right),$$

where R_0^* is parameterized in (61).

4 Equivalent reduction of precedence relation systems

This section presents a full parameterization of the set of all equivalent reductions of any precedence relation system. The parameterization will indicate that contrary to the problem of finding the maximum redundant edge set which is NP-hard, every equivalent reduction can be computed in polynomial-time. In the following, we first present some preparatory results in Section 4.1. Next, the main result on the parameterization of equivalent reduction is described in Section 4.2. Section 4.3 describes a further simplification of equivalent reduction, which is connected to the maximum redundant edge set of the condensation of the original precedence graph.

Because of the correspondence between a precedence relation system and its precedence graph, in this section we shall extend the notion of equivalent reduction in Definition 2 to precedence graphs. Given a precedence graph G , we call another precedence graph G' equivalent to G , with notation $G' \equiv G$, if the precedence relation systems corresponding to G and G' are equivalent (i.e., they have the same solution set). Consequently, an equivalent reduction of a precedence graph G is a precedence graph $G' \equiv G$ such that G' has the minimum possible number of edges.

4.1 Preparatory results

The following statements will be used in the proof of the main result to establish some graph-based necessary conditions for the equivalent reductions of a precedence graph.

Lemma 20. *Let $G = (V, E, c(E))$ and $G' = (V, E', c'(E'))$ be two equivalent precedence graphs (i.e., $G \equiv G'$). Then, if $(f, g) \in E$ with weight c_{fg} , then there exists a path $p'_{fg} : f \rightsquigarrow g$ in G' with weight $c'_{p'_{fg}} \leq c_{fg}$.*

Proof. Corresponding to $(f, g) \in E$ is an inequality $x_f - x_g \leq c_{fg}$ in G . Since $G' \equiv G$, whenever $x \in \mathbb{R}^n$ satisfies G' , x also satisfies $x_f - x_g \leq c_{fg}$. This is condition Lemma 2.a applied to G' . Hence, Lemma 2 implies that in G' there exists a path $p'_{fg} : f \rightsquigarrow g$ with weight $c'_{p'_{fg}} \leq c_{fg}$. \square

Remark 10. Lemma 20 can in fact be extended (the details omitted) to show that the equivalent reduction problem has the following graph interpretation: given precedence graph $G = (V, E, c(E))$, find (possibly another) precedence graph $G' = (V, E', c'(E'))$ satisfying

1. For each $(i, j) \in E$ with weight c_{ij} there exists a path $p'_{ij} : i \rightsquigarrow j$ in G' with weight $c'_{p'_{ij}} \leq c_{ij}$. Conversely, for each $(u, v) \in E'$ with weight c'_{uv} there exists a path $p_{uv} : u \rightsquigarrow v$ in G with weight $c_{p_{uv}} \leq c'_{uv}$,
2. with respect to the first bullet, E' has minimum cardinality.

From this graph interpretation it is also possible to see that the equivalent reduction problem is a generalization of the transitive reduction problem for unweighted directed graphs in [2]. That is, an instance of transitive reduction problem is an instance of equivalent reduction problem with edge weights c_{ij} set to zero or a negative constant.

Lemma 21. *Let $G = (V, E, c(E))$ and $G' = (V, E', c'(E'))$ be two equivalent precedence graphs (i.e., $G \equiv G'$). Then, for $i, j \in V$, $i \sim j$ in G if and only if $i \sim j$ in G' .*

Proof. The statement holds trivially when $i = j$. Therefore, only the cases when $i \neq j$ are considered. Suppose $i \sim j$ in G . Then there exists a zero-weight closed walk $w_{iji} : (i = i_0, i_1, \dots, i_k = j, i_{k+1}, \dots, i_m = i)$ in G . Since Lemma 20 and Lemma 21 have the same statement assumption, Lemma 20 implies that for each $q = 0, 1, \dots, m - 1$ there exists a path $p'_{i_q i_{q+1}} : i_q \rightsquigarrow i_{q+1}$ in G' whose path weight is less than or equal to the weight of the corresponding edge (i_q, i_{q+1}) in G . Hence, in G' there is a closed walk $w'_{iji} : i \rightsquigarrow j \rightsquigarrow i$ with weight less than or equal to that of w_{iji} (which is zero). Since G' is a precedence graph, Assumption 1.b implies that the weight of w'_{iji} cannot be negative. Hence, $i \sim j$ in G' . The above argument can be repeated, with appropriate modifications, to show that $i \sim j$ in G , whenever $i \sim j$ in G' . Hence, the desired statement is established. \square

Remark 11. Lemma 8 is in fact a corollary of Lemma 21.

Lemma 22. *Let $G = (V, E, c(E))$ be a precedence graph. Let $G^* = (V, E^*, c(E^*))$ with $E^* \subseteq E$, and $G' = (V, E', c'(E'))$ with $E' \subseteq V \times V$ and $c'(E') \in \mathbb{R}^{|E'|}$ be given. Assume that*

- $G \equiv G^* \equiv G'$.
- Neither G^* nor G' has any redundant edge (cf. Definition 4).
- For $k \in \{1, 2, \dots, K\}$, $[v_k]$ is the equivalence class induced by relation \sim in G , G^* and G' (the three precedence graphs having the same equivalence class partitioning is a consequence of Lemma 21 with $G \equiv G^* \equiv G'$).

- For $i, j \in V$ and $i \neq j$, let $E_{ij}^* := E^* \cap ([v_i] \times [v_j])$, and $E'_{ij} := E' \cap ([v_i] \times [v_j])$. That is, E_{ij}^* is the subset of E^* whose member edges go from $[v_i]$ to $[v_j]$, and E'_{ij} is defined analogously.

Then,

$$E_{ij}^* = \emptyset \iff E'_{ij} = \emptyset, \quad \forall 1 \leq i \neq j \leq K. \quad (64)$$

In addition, suppose $(f, g) \in E_{ij}^*$ with weight c_{fg} and $(u, v) \in E'_{ij}$ with weight c'_{uv} . Then,

$$c'_{uv} = d_{uf} + c_{fg} + d_{gv}, \quad (65)$$

with d_{uf} and d_{gv} being the minimum walk weights in G defined in Definition 6.

Proof. To show (64) by contradiction, first assume that $E_{ij}^* = \emptyset$ but $E'_{ij} \neq \emptyset$. Let $(u, v) \in E'_{ij}$ with weight c'_{uv} . Since $G^* \equiv G'$, Lemma 20 guarantees the existence of a path $p_{uv} : u \rightsquigarrow v$ in G^* such that $c_{p_{uv}} \leq c'_{uv}$. Further, since $E_{ij}^* = \emptyset$, p_{uv} must traverse a node $t \notin ([v_i] \cup [v_j])$ and hence it is of the form $p_{uv} : u \rightsquigarrow t \rightsquigarrow v$. Applying Lemma 20 to each edge in p_{uv} results in a walk $w'_{uv} : u \rightsquigarrow t \rightsquigarrow v$ in G' such that $c'_{w'_{uv}} \leq c_{p_{uv}} \leq c'_{uv}$. If (u, v) is not part of w'_{uv} , then (u, v) is a redundant edge in G' . This is a contradiction. Thus, (u, v) is part of w'_{uv} , implying that w'_{uv} is either (a) $u \rightsquigarrow u \rightarrow v \rightsquigarrow t \rightsquigarrow v$ or (b) $u \rightsquigarrow t \rightsquigarrow u \rightarrow v \rightsquigarrow v$. In the case of (a), we distinguish two cases:

- the weight of the closed walk $v \rightsquigarrow t \rightsquigarrow v$ is zero (G' cannot have negative weight closed walk because of Assumption 1.b). Then, $t \sim v$ in G' , and this is a contradiction since $t \notin [v_j]$.
- the weight of the closed walk $v \rightsquigarrow t \rightsquigarrow v$ is positive. Then, $c'_{w'_{uv}} \leq c'_{uv}$ implies that the weight of the closed walk $u \rightsquigarrow u$ is negative. This is also a contradiction of Assumption 1.b.

A similar argument can show that case (b) also leads to contradictory conclusions. Therefore, the original assumption that $E_{ij}^* = \emptyset$ but $E'_{ij} \neq \emptyset$ cannot hold. Further, an analogous argument can show that $E'_{ij} = \emptyset$ but $E_{ij}^* \neq \emptyset$ cannot hold, and hence (64) is established.

Now we show (65) by contradiction. First, assume that

$$c'_{uv} > d_{uf} + c_{fg} + d_{gv}. \quad (66)$$

Because $E^* \subseteq E$, G^* is a subgraph of G . In addition, since $u, f \in [v_i]$, $g, v \in [v_j]$, $[v_i]$ and $[v_j]$ are equivalence classes of G^* (as argued in the statement), Lemma 10.a specifies that there are two walks $u \rightsquigarrow f$ and $g \rightsquigarrow v$

in G^* with weights d_{uf} and d_{gv} respectively. Therefore, the right-hand side of (66) is the weight of a walk $w_{uv} : u \rightsquigarrow f \rightarrow g \rightsquigarrow v$ in G^* . Applying Lemma 20 to each edge of w_{uv} yields a walk $w'_{uv} : u \rightsquigarrow f \rightsquigarrow g \rightsquigarrow v$ in G' such that

$$c'_{uv} > d_{uf} + c_{fg} + d_{gv} \geq c'_{w'_{uv}}. \quad (67)$$

If (u, v) is not part of w'_{uv} then (67) implies that (u, v) is a redundant edge in G' . This contradicts the statement assumption. On the other hand, if (u, v) is part of w'_{uv} , then w'_{uv} is of the form $u \rightsquigarrow u \rightarrow v \rightsquigarrow v$. (67) implies that $w'_{uv} \neq (u, v)$ and at least one of the closed walks $u \rightsquigarrow u$ and $v \rightsquigarrow v$ have negative weight. This contradicts Assumption 1.b. In conclusion, (66) cannot hold. Next, we assume that

$$c'_{uv} < d_{uf} + c_{fg} + d_{gv}. \quad (68)$$

By Lemma 10.b, $d_{uf} = -d_{fu}$ and $d_{gv} = -d_{vg}$. Hence, (68) is equivalent to

$$c_{fg} > d_{fu} + c'_{uv} + d_{vg}. \quad (69)$$

In addition, Lemma 20 applied to $(u, v) \in E'_{ij}$ yields a walk $w_{uv} : u \rightsquigarrow v$ in G^* such that $c_{w_{uv}} \leq c'_{uv}$. This, together with (69), implies

$$c_{fg} > d_{fu} + c_{w_{uv}} + d_{vg}. \quad (70)$$

By a similar argument as in (66), Lemma 10.a guarantees the existence of two walks $f \rightsquigarrow u$ and $v \rightsquigarrow g$ in G^* with weights d_{fu} and d_{vg} respectively. Consequently, in G^* there exists a walk $w_{fg} : f \rightsquigarrow g$ such that $c_{w_{fg}} < c_{fg}$. If (f, g) is not part of w_{fg} then (70) implies that (f, g) is a redundant edge in G^* . This contradicts the statement assumption. On the other hand, if (f, g) is part of w_{fg} , then w_{fg} is of the form $f \rightsquigarrow f \rightarrow g \rightsquigarrow g$. (70) implies that $w_{fg} \neq (f, g)$ and at least one of the closed walks $f \rightsquigarrow f$ and $g \rightsquigarrow g$ have negative weight. This again contradicts Assumption 1.b. Hence, (68) does not hold. Consequently, (65) must hold. \square

4.2 Parameterization of equivalent reduction

Analogous to the maximum redundant edge set problem, an equivalent reduction can be decomposed into $K+1$ components. However, all components of equivalent reduction can be computed in polynomial-time. The following statement summarizes the decomposition result related to equivalent reduction:

Theorem 4. Let $G = (V, E, c(E))$ be a precedence graph. Let the following be defined in the context of G :

- K is the number of equivalence classes induced by relation \sim in Definition 7.a.
- For $k \in \{1, 2, \dots, K\}$, $[v_k]$ denotes the equivalence class defined in Definition 7.b, with v_k being the representing node of $[v_k]$.
- For $i, j \in V$, d_{ij} is the minimum walk weight defined in Definition 6.
- For $1 \leq i \neq j \leq K$, E_{ij} and (v_{ij}^i, v_{ij}^j) are defined in Definition 7.d.
- \tilde{G} is the condensation of G defined in Definition 8, in accordance with the designation of representing nodes v_k 's.

Then, every equivalent reduction of G (defined in Definition 2) can be parameterized by

$$(V, E^{er}, c^{er}(E^{er})). \quad (71)$$

In (71), the edge set E^{er} is parameterized by

$$E^{er} = E_0^{er} \cup \left(\bigcup_{k=1}^K E_k^{er} \right),$$

where

for $k \in \{1, \dots, K\}$, $E_k^{er} = \emptyset$ if $|[v_k]| = 1$, otherwise E_k^{er} contains $|[v_k]|$ edges forming a zero weight directed cycle traversing all nodes in $[v_k]$. (72)

In addition, E_0^{er} can be decomposed into

$$E_0^{er} = \bigcup_{1 \leq i \neq j \leq K} E_{ij}^{er},$$

where

$$E_{ij}^{er} = \begin{cases} (u, v) \in [v_i] \times [v_j], & \text{if } E_{ij} \neq \emptyset \text{ and } (v_i, v_j) \notin \tilde{R}_0^* \\ \emptyset, & \text{if } E_{ij} = \emptyset \text{ or } (v_i, v_j) \in \tilde{R}_0^* \end{cases}, \quad (73)$$

and \tilde{R}_0^* is the maximum redundant edge set of \tilde{G} , the condensation of G . In (71), the edge weights $c^{er}(E^{er})$ are defined by

$$c_{uv}^{er} = \begin{cases} d_{uv_{ij}^i} + c_{v_{ij}^i, v_{ij}^j} + d_{v_{ij}^j, v}, & \text{if } (u, v) \in [v_i] \times [v_j], i \neq j, \\ d_{uv}, & \text{if } (u, v) \in [v_i] \times [v_i]. \end{cases} \quad (74)$$

Remark 12. It appears that the statement of Theorem 4 may depend on some arbitrary choices. For instance, \tilde{R}_0^* may depend on the choices of representing nodes v_1, v_2, \dots, v_K for the equivalence classes, and the designation of (v_{ij}^i, v_{ij}^j) in (73) is arbitrary (see Definition 7.d). However, it turns out that Theorem 4 is independent of arbitrary choices. In particular, we can show (in Appendix C) that

- As in the case of Remark 8, for all $1 \leq i \neq j \leq K$, $|\tilde{R}_0^* \cap ([v_i] \times [v_j])|$ (which can only be 0 or 1) is independent of the choices of v_1, \dots, v_K .
- In (74), c_{uv}^{er} is independent of the designation of (v_{ij}^i, v_{ij}^j) because

$$d_{uv_{ij}^i} + c_{v_{ij}^i, v_{ij}^j} + d_{v_{ij}^j, v} = \min_{(s,t) \in E_{ij}} d_{us} + c_{st} + d_{tv}.$$

The main difference between the computations in Theorem 3 and Theorem 4 is that in an equivalent reduction, the edges in each equivalence class with more than one node form a zero-weight cycle. The formation of these cycles requires only $O(|[v_k]|)$ time for each equivalence class $[v_k]$. Hence, forming all cycles for all equivalent classes requires only $O(|V|)$ time. This difference is analogous to the distinction between the solutions to minimum equivalent graph problem in [1] and transitive reduction problem in [2]. Figure 7 shows two equivalent reductions of the example graph in Figure 3.

Proof of Theorem 4. The proof is divided into three parts. In the first part, some necessary conditions of equivalent reduction are listed. In the second part, an alternative characterization of the set of all equivalent reductions is introduced. Finally, in the third part it is established that the alternative characterization is in fact the parameterization provided in the statement of Theorem 4. To begin with the first part, let

$$G^* = (V, E^*, c(E^*)) := (V, E \setminus R^*, c(E \setminus R^*)), \quad (75)$$

with R^* being a maximum redundant edge set of G parameterized in Theorem 3, where in (62) the edge (g, h) is chosen to be (v_{ij}^i, v_{ij}^j) for $1 \leq i \neq j \leq K$. Then, because R^* is a redundant edge set Definition 3, Lemma 4 and Definition 1 imply that $G^* \equiv G$. In addition, by definition of maximum redundant edge set G^* does not have any redundant edge. Further, by (62)

$$E^* \cap ([v_i] \times [v_j]) = \begin{cases} (v_{ij}^i, v_{ij}^j), & \text{if } E_{ij} \neq \emptyset \text{ and } (v_i, v_j) \notin \tilde{R}_0^*, \\ \emptyset, & \text{if } E_{ij} = \emptyset \text{ or } (v_i, v_j) \in \tilde{R}_0^*, \end{cases} \quad (76)$$

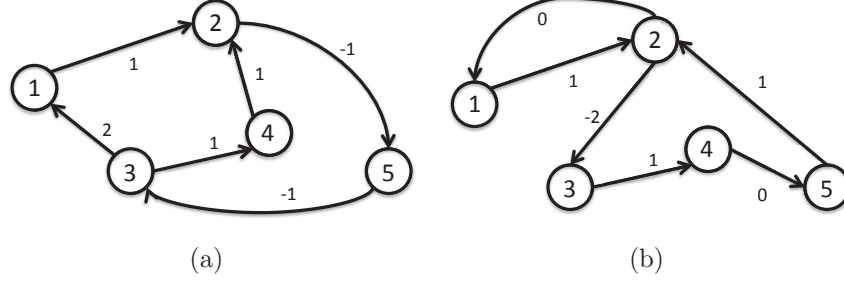


Figure 7: Two equivalent reductions of the graph in Figure 3. Figure 7a is the graph in Figure 3 without edge $(3, 2)$ since it is not needed to form the zero-weight cycle traversing nodes 2,3,4,5. It can be verified that Figure 7a satisfies the parameterization in (71). On the other hand, while Figure 7b resembles less the graph in Figure 3, it is in fact another equivalent reduction. The cycle $(2, 3, 4, 5, 2)$ and the corresponding edge weights satisfy (72) and (74). In addition, by (73) there should be one edge each from $[1]$ to $[2]$ and from $[2]$ to $[1]$, since $E_{12} \neq \emptyset$, $E_{21} \neq \emptyset$ and $\tilde{R}_0^* = \emptyset$. The edges $(1, 2)$ and $(2, 1)$ together with their weights satisfy (73) and (74).

for $1 \leq i \neq j \leq K$. Now we analyze the properties of equivalent reduction. Let \mathcal{G}^{ER} denote the set of all equivalent reductions of G . That is,

$$\mathcal{G}^{ER} = \underset{G'=(V,E',c'(E')) \equiv G}{\operatorname{argmin}} |E'|. \quad (77)$$

Let $G' = (V, E', c'(E')) \in \mathcal{G}^{ER}$. In the following, three properties of E' will be discussed regarding

- self-loops and parallel edges,
- edges between two different equivalence classes,
- edges between two different nodes in an equivalence class.

First, we claim that E' does not contain any self-loops or parallel edges. That is,

$$(i, i) \notin E', \quad \forall i \in V, \quad \text{and all elements of } E' \text{ are distinct.} \quad (78)$$

To see (78), first note that G' has nonempty solution set because $G' \equiv G$ and G has nonempty solution set because of Assumption 1.a. Hence, the weight c'_{ii} of any self-loop (i, i) must be nonnegative, and the corresponding

precedence inequality $x_i - x_i = 0 \leq c'_{ij}$ is redundant and should not be present in $G' \in \mathcal{G}^{ER}$. It is also clear that if G' has the minimum number of edges as characterized by (77), it is impossible to have more than one edge connecting two nodes.

Second, for the edges between equivalent classes, note that $G' \in \mathcal{G}^{ER}$ implies $G' \equiv G$ and G' does not have any redundant edge. Hence, by applying Lemma 22 with G' and G^* , it can be concluded from (64) and (76) that

$$E' \cap ([v_i] \times [v_j]) = \begin{cases} (u, v) \in [v_i] \times [v_j], & \text{if } E_{ij} \neq \emptyset \text{ and } (v_i, v_j) \notin \tilde{R}_0^* \\ \emptyset, & \text{if } E_{ij} = \emptyset \text{ or } (v_i, v_j) \in \tilde{R}_0^* \end{cases}, \quad (79)$$

for $1 \leq i \neq j \leq K$. In addition, by (65) the edge weight in (79) is

$$c'_{uv} = d_{uv_{ij}^i} + c_{v_{ij}^i, v_{ij}^j} + d_{v_{ij}^j, v}. \quad (80)$$

(79) implies that

$$\left| E' \cap \left(\bigcup_{1 \leq i \neq j \leq K} [v_i] \times [v_j] \right) \right| = |\tilde{E}_0| - |\tilde{R}_0^*|, \quad (81)$$

where \tilde{E}_0 is the edge set of any condensation of G .

Third, we consider the edges in equivalence class $[v_k]$ where $|[v_k]| \geq 2$. Since $G' \equiv G$, Lemma 21 states that

$$\forall i, j \in V, \quad i \sim j \text{ in } G' \iff i \sim j \text{ in } G. \quad (82)$$

Condition (82) implies that for $k \in \{1, 2, \dots, K\}$,

$$\text{The graph } H_k := ([v_k], E' \cap ([v_k] \times [v_k])) \text{ is connected,} \quad (83a)$$

$$|[v_k]| \geq 2 \implies \text{every node in } H_k \text{ is incident to at least two edges,} \quad (83b)$$

$$|[v_k]| \geq 2 \implies H_k \text{ cannot be a tree.} \quad (83c)$$

Thus, $G' \in \mathcal{G}^{ER}$ implies (83), which in turn implies that

$$|[v_k]| \geq 2 \implies |E' \cap ([v_k] \times [v_k])| \geq |[v_k]|, \quad (84)$$

because if a connected graph has more than one node and it is not a tree, then it has at least as many edges as the number of nodes (e.g., [24]). This concludes the first part of the proof (of Theorem 4), listing of properties of $G' \in \mathcal{G}^{ER}$.

For the rest of the proof, G' is not assumed to be a member of \mathcal{G}^{ER} . We begin the second part of the proof by considering a (to be shown to be) alternative description of \mathcal{G}^{ER} . Denote the set

$$\mathcal{G}^* := \{G' = (V, E', c'(E')) \mid G' \equiv G, G' \text{ satisfies (78), (79), (80), (86)}\}, \quad (85)$$

where condition (86) is defined as

$$|[v_k]| \geq 2 \implies |E' \cap ([v_k] \times [v_k])| = |[v_k]|. \quad (86)$$

In essence, \mathcal{G}^* in (85) is the (to be shown to be nonempty) subset of the feasible set in the optimization problem in (77), whose elements attain the lower bound of the total number of edges of G' specified by (78), (81) and (84). In other words, (77), (78), (81) and (84) together imply that

$$\mathcal{G}^* \neq \emptyset \implies \mathcal{G}^* = \mathcal{G}^{ER}. \quad (87)$$

The set \mathcal{G}^* in (85) can be described in a more convenient form. Its derivation is based on the establishment of two properties of \mathcal{G}^* to be shown in (88) and (90). As (88), it is claimed that if $G' \in \mathcal{G}^*$ then for all $k \in \{1, 2, \dots, K\}$,

$$\begin{aligned} |[v_k]| = 1 &\implies E' \cap ([v_k] \times [v_k]) = \emptyset, \\ |[v_k]| \geq 2 &\implies E' \cap ([v_k] \times [v_k]) \text{ forms a zero weight directed cycle} \\ &\text{with } |[v_k]| \text{ edges traversing all nodes in } [v_k]. \end{aligned} \quad (88)$$

The first implication in (88) is a specialization of (78). For the second implication in (88), note that $G' \in \mathcal{G}^*$ satisfies (86). Hence, restricted to the graph $H_k := ([v_k], E' \cap ([v_k] \times [v_k]))$ the sum of degrees (in-degrees and out-degrees together) of all nodes is $2 \times |[v_k]|$ (e.g., [25, Theorem 15.3]). If in H_k there is a node with degree greater than two, then there exists at least one other node in H_k with degree less than two. This violates (83b), and hence conditions (82) and $G' \equiv G$ are violated as well. Thus, every node in H_k has degree exactly two. This, together with the no self-loop condition in (78), implies that

$$|[v_k]| \geq 2 \implies \text{every node in } H_k \text{ is incident to exactly two edges.} \quad (89)$$

Note that $G' \equiv G$ since $G' \in \mathcal{G}^*$. Hence, G' satisfies (82) and (83a). Then, it is claimed that (82), (83a) and (89) together imply the second implication in (88). (82) and (89) imply that for each node in $[v_k]$ there is one outgoing edge and one incoming edge in H_k . This, together with the fact that H_k is a finite

graph, suggests that starting from any node $i_0 \in [v_k]$ and following the edges along their directions it is possible to trace a cycle $w = (i_0, i_1, \dots, i_m = i_0)$ in H_k . If there exists $i_{m+1} \in [v_k]$ that is not part of cycle w , then by following the outgoing edge of i_{m+1} and subsequent edges along their directions either one of the following is resulted: (i) another cycle w' which is disjointed from w is traced, or (ii) there exists $0 < q \leq m$ such that i_q is connected to some $v \in [v_k]$, v not part of w . Case (i) contradicts (83a). On the other hand, case (ii) implies that i_q is connected to i_{q-1} , $i_{((q+1) \bmod m)}$ and v . This suggests that i_q has degree three in H_k , and contradicts (89). Thus, both (i) and (ii) lead to contradictory conclusions, and hence w is a directed cycle traversing all nodes in $[v_k]$. Finally, the fact that w is a zero weight cycle is due to (82). Thus, the second implication in (88) is established. Furthermore, as a consequence of $G' \in \mathcal{G}^*$ (in particular $G' \equiv G$ and (88)), it is claimed that

$$(u, v) \in E' \cap ([v_k] \times [v_k]) \text{ for some } k \implies c'_{uv} = d_{uv}, \quad (90)$$

where we note that d_{uv} is the minimum walk weight from u to v in G (Definition 6). The proof of (90) is as follows: let $(i_0, i_1, \dots, i_m = i_0)$ be the directed cycle in $[v_k]$ in (88). Then, (90) holds if it is true that

$$c'_{i_q i_{q+1}} = d_{i_q i_{q+1}}, \quad \forall q = 0, 1, \dots, m-1. \quad (91)$$

The proof of (91) is as follows: since $G' \equiv G$, by Lemma 20 for each $q = 0, 1, \dots, m-1$ there exists a walk $w_{i_q i_{q+1}} : i_q \rightsquigarrow i_{q+1}$ in G such that $c_{w_{i_q i_{q+1}}} \leq c'_{i_q i_{q+1}}$. In addition, since $d_{i_q i_{q+1}} \leq c_{w_{i_q i_{q+1}}}$ by Definition 6, it holds that

$$\forall q = 0, \dots, m-1, \quad d_{i_q i_{q+1}} \leq c_{w_{i_q i_{q+1}}} \leq c'_{i_q i_{q+1}} \implies c'_{i_q i_{q+1}} - d_{i_q i_{q+1}} \geq 0. \quad (92)$$

In addition, by Lemma 10.b and 10.c

$$d_{i_0 i_1} + d_{i_1 i_2} + \dots + d_{i_{m-1} i_m} = d_{i_0 i_{m-1}} + d_{i_{m-1} i_m} = d_{i_0 i_{m-1}} - d_{i_0 i_{m-1}} = 0. \quad (93)$$

Since $(i_0, i_1, \dots, i_m = i_0)$ is a zero weight cycle, it holds that

$$c'_{i_0 i_1} + c'_{i_1 i_2} + \dots + c'_{i_{m-1} i_m} = 0. \quad (94)$$

Combining (93) and (94), it can be concluded that

$$(c'_{i_0 i_1} - d_{i_0 i_1}) + (c'_{i_1 i_2} - d_{i_1 i_2}) + \dots + (c'_{i_{m-1} i_m} - d_{i_{m-1} i_m}) = 0. \quad (95)$$

Then, (92) and (95) together lead to (91). In summary,

$$G' \in \mathcal{G}^* \implies G' \equiv G, G' \text{ satisfies (78), (79), (80), (88), (90).}$$

Since (88) leads to (86), it holds that

$$\mathcal{G}^* = \{G' \mid G' \equiv G, G' \text{ satisfies (78), (79), (80), (88), (90)}\}. \quad (96)$$

This concludes the second part of the proof (of Theorem 4), defining and characterizing \mathcal{G}^* as a possible alternative description of \mathcal{G}^{ER} , the set of all equivalent reductions of G .

In the last part of the proof, we establish the connection between \mathcal{G}^* and the parameterization in the statement of Theorem 4. Denote \mathcal{G}^{er} as the set of all precedence graphs satisfying (71) to (74) (i.e., the parameterization in the statement of Theorem 4). Then, it can be seen that

$$\mathcal{G}^{er} = \{G' \mid G' \text{ satisfies (78), (79), (80), (88), (90)}\}. \quad (97)$$

That is, the set \mathcal{G}^{er} in (97) includes \mathcal{G}^* in (96), since the definition of the \mathcal{G}^{er} is the same as \mathcal{G}^* except that the requirement $G' \equiv G$ is removed. As a side note, $G' \in \mathcal{G}^{er}$ is the only precedence graph considered in this paper where Assumption 1.a (i.e., feasibility) cannot be taken for granted because the condition $G' \equiv G$ has not been shown. In the remaining part of the proof we will show that

$$G' = (V, E', c'(E')) \in \mathcal{G}^{er} \implies G' \equiv G^* (\equiv G), \quad (98)$$

where G^* is defined in the earlier part of the proof in (75). If (98) holds, then the desired statement in the theorem is shown because

$$\mathcal{G}^{er} = \mathcal{G}^* \implies \mathcal{G}^* \neq \emptyset \text{ (since } \mathcal{G}^{er} \neq \emptyset \text{)} \xrightarrow{(87)} \mathcal{G}^{er} = \mathcal{G}^* = \mathcal{G}^{ER}.$$

Now we start to show (98), and assume that $G' \in \mathcal{G}^{er}$. We first show one direction of “ \equiv ” in (98) by proving

$$x \in \mathbb{R}^n \text{ satisfies } G^* \implies x_u - x_v \leq c'_{uv}, \quad \forall (u, v) \in E'. \quad (99)$$

There are two cases in (99): (a) $u \neq v, u, v \in [v_k]$ for some k , or (b) $u \in [v_k]$ and $v \in [v_q]$ with $k \neq q$. For case (a), (90) requires that the weight of (u, v) is $c'_{uv} = d_{uv}$. Since $G^* \equiv G$, Lemma 21 implies that G^* and G have the same equivalence class partitioning. Thus, by Lemma 10.a there exists a walk $u \rightsquigarrow v$ in G^* with weight d_{uv} , which is the same as the weight of (u, v) in G' . Then, Lemma 2 (applied to G^*) establishes (99) in case (a). For case (b), by (76) and (79) $(u, v) \in E'$ implies that $(v_{kq}^k, v_{kq}^q) \in E^*$. In addition, (80) implies that $c'_{uv} = d_{uv_{kq}^k} + c_{v_{kq}^k v_{kq}^q} + d_{v_{kq}^q v}$. Since $u, v_{kq}^k \in [v_k]$, $v, v_{kq}^q \in [v_q]$, Lemma 10.a specifies that there exists two walks in G^* , $u \rightsquigarrow v_{kq}^k$

and $v_{kq}^q \rightsquigarrow v$, with weights $d_{uv_{kq}^k}$ and $d_{v_{kq}^q v}$ respectively. Hence, in G^* there exists a walk $u \rightsquigarrow v_{kq}^k \rightarrow v_{kq}^q \rightsquigarrow v$ with weight equal to c'_{uv} . Again, by Lemma 2 it can be concluded that (99) holds under case (b). In conclusion, (99) holds. Next, we show the other direction of “ \equiv ” in (98) by establishing

$$x \in \mathbb{R}^n \text{ satisfies } G' \implies x_i - x_j \leq c_{ij}, \quad \forall (i, j) \in E^*. \quad (100)$$

First, we note that (99) implies that G' has at least one solution since G^* has at least one. Therefore, G' also satisfies Assumption 1 and Lemma 2 can be applied to G' . There are two cases in (100): (c) $i \neq j, i, j \in [v_k]$ for some k , or (d) $i \in [v_k]$ and $j \in [v_q]$ with $k \neq q$. For case (c), by (88) and (90) there exists a path $(i = i_0, i_1, \dots, i_m = j)$ in G' , as a part of the cycle traversing nodes in $[v_k]$, with weight $d_{i_0 i_1} + \dots + d_{i_{m-1} i_m} = d_{ij} \leq c_{ij}$ (the equality is due to Lemma 10.c and the inequality is due to Definition 6). Thus, by Lemma 2 (100) holds in case (c). For case (d), (76), (79) and (80) specify that $i = v_{kq}^k, j = v_{kq}^q$ and there exists $(u, v) \in E'$ with $u \in [v_k], v \in [v_q]$ such that $c_{ij} = d_{iu} + c'_{uv} + d_{vj}$. Since $i, u \in [v_k]$ and $v, j \in [v_q]$, (88) and (90) imply the existence of two paths $i \rightsquigarrow u$ and $v \rightsquigarrow j$ in G' with weights d_{iu} and d_{vj} respectively. Hence, in G' there exists a path $i \rightsquigarrow u \rightarrow v \rightsquigarrow j$ with weight $d_{iu} + c'_{uv} + d_{vj} = c_{ij}$. Consequently, by Lemma 2 (100) holds in case (d). Therefore, (100) holds and consequently (98) holds. This concludes the proof of Theorem 4. \square

4.3 Variable elimination in equivalent reduction

Let $G^{er} = (V, E^{er}, c^{er}(E^{er}))$ denote an equivalent reduction of a precedence graph G . By Theorem 4 (i.e., (72) and (74)) in G^{er} if $i, j \in [v_k]$ for some k then $x_i - x_j = d_{ij}$ for all $x \in \mathbb{R}^n$ satisfying G . In some applications (e.g., optimization), it is beneficial to use these equalities to eliminate all except one variable in each equivalence class (e.g., keeping only the representing node v_k). The resulted simplified inequality system can be considered as a “condensation” of G^{er} constructed as follows:

Algorithm 3 (Condensation of equivalent reduction).

1. Designate representing nodes in the equivalence classes as v_1, v_2, \dots, v_K .
2. Define $\tilde{V} := \{v_1, v_2, \dots, v_K\}$.
3. Define $\hat{E}_0^{er} := \{(v_i, v_j) \mid E^{er} \cap ([v_i] \times [v_j]) = E_{ij}^{er} \neq \emptyset, i \neq j\}$.

4. For each $(v_i, v_j) \in \hat{E}_0^{er}$, define edge weight

$$\hat{c}_{v_i v_j}^{er} := d_{v_i u} + c_{uv}^{er} + d_{vv_j}, \quad (101)$$

where $\{(u, v)\} = E^{er} \cap ([v_i] \times [v_j])$.

5. Define the condensation of G^{er} to be $(\tilde{V}, \tilde{E}_0^{er}, \hat{c}^{er}(\tilde{E}_0^{er}))$.

It turns out that the condensation of an equivalent reduction of G can be interpreted in an alternative way related to the condensation of G :

Lemma 23. *Let $G^{er} = (V, E^{er}, c^{er}(E^{er}))$ be an equivalent reduction of a precedence graph G . Let \hat{E}_0^{er} and $\hat{c}^{er}(\hat{E}_0^{er})$, defined in Algorithm 3, describe the condensation of G^{er} . Let $\tilde{G} = (\tilde{V}, \tilde{E}_0, \tilde{c}(\tilde{E}_0))$ denote the condensation of G , with the same designation of v_1, v_2, \dots, v_K as in the choice of Algorithm 3. Let \tilde{R}_0^* be the maximum redundant edge set of \tilde{G} . Then,*

$$(\tilde{V}, \hat{E}_0^{er}, \hat{c}^{er}(\hat{E}_0^{er})) = (\tilde{V}, \tilde{E}_0 \setminus \tilde{R}_0^*, \tilde{c}(\tilde{E}_0 \setminus \tilde{R}_0^*)). \quad (102)$$

That is, the condensation of an equivalent reduction of G is the condensation of G with its maximum redundant edge set removed.

Proof. To see (102), we compare the following:

- \tilde{V} is the node set in the left-hand and right-hand sides of (102).
- For the edge sets, by the definition in Algorithm 3 and Theorem 4 (i.e., (73)) it holds that

$$(v_i, v_j) \in \hat{E}_0^{er} \iff E_{ij}^{er} \neq \emptyset \iff (v_i, v_j) \in (\tilde{E}_0 \setminus \tilde{R}_0^*).$$

Hence, the left-hand and right-hand sides of (102) have the same edge sets (i.e., $\hat{E}_0^{er} = \tilde{E}_0 \setminus \tilde{R}_0^*$).

- For each $(v_i, v_j) \in \hat{E}_0^{er}$, the edge weight satisfies

$$\begin{aligned} \hat{c}_{v_i v_j}^{er} &\stackrel{(101)}{=} d_{v_i u} + c_{uv}^{er} + d_{vv_j} \\ &\stackrel{(74)}{=} d_{v_i u} + d_{uv_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v} + d_{vv_j} \\ &\stackrel{\text{Lemma 10.c}}{=} d_{v_i v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v_j} \\ &\stackrel{(28)}{=} \tilde{c}_{v_i v_j} \end{aligned}$$

□

5 Conclusions

The maximum index set of redundant relations problem (i.e., maximum redundant edge set problem) is a generalization of the minimum equivalent graph problem. Similarly, the equivalent reduction problem is a generalization of the transitive reduction problem. Nevertheless, the generalizations are shown to possess analogous computation properties of the respective restrictions. The maximum redundant edge set problem is NP-hard in general, and it is solvable in polynomial time if the graph does not have zero-weight cycles. This is analogous to the statement that the minimum equivalent graph problem is NP-hard in general, and it is solvable in polynomial time if the graph is acyclic. In addition, the decomposition of the maximum redundant edge set problem based on the equivalence classes defined by the “on-a-zero-weight-closed-walk” relation is analogous to the decomposition of the minimum equivalent graph problem based on strongly connected components. Further, in the decomposition, the subproblems dealing with the edges between equivalence classes are both solvable in polynomial time. The subproblem within an equivalence class for maximum redundant edge set problem is in fact equivalent to the (NP-hard) minimum equivalent graph problem within the corresponding equivalence class, with the implication that all available exact or inexact algorithms for minimum equivalent graph problem can be utilized. Finally, analogous results also hold between the equivalent reduction problem and the transitive reduction problem. The structure of decompositions of the solutions are analogous, and both problems can be solved in polynomial-time using similar shortest path calculations.

A Decomposing a walk into a path and cycles

In a directed graph $G = (V, E)$ without parallel edges, a walk can be represented by a sequence of node indices $w = (i_0, i_1, \dots, i_m)$ where $(i_k, i_{k+1}) \in E$ for $k = 0, 1, \dots, m - 1$. To decompose a walk w , the following “scan” operation is necessary:

$S = \text{scan}(w)$

- Initialize $S \leftarrow \emptyset$ and $k \leftarrow 0$.
- While $k < \text{length}(w)$, do

1. If there exists r as the smallest index such that $r > k$ and $i_r = i_k$, then update $S \leftarrow S \cup \{(i_k, i_{k+1}, \dots, i_r)\}$. The sequence w is also updated according to

$$w \leftarrow \begin{cases} () & \text{if } k = 0 \text{ and } r = m \\ (i_0, i_1, \dots, i_k, i_{r+1}, i_{r+2}, \dots, i_m) & \text{otherwise} \end{cases}.$$

In updating w , the sub-walk (i_{r+1}, \dots, i_m) is empty by convention if $r = m$.

2. Increase $k \leftarrow k + 1$.

End (of While)

- Update $S \leftarrow S \cup \{w\}$.

For any two nodes u and v , applying the scan operation to a walk $w = (u = i_0, i_1, \dots, i_m = v)$ results in S a path from u to v (can be a degenerate path containing a single node if $u = v$) and a finite number of closed walks (if any). The scan operation can be applied to each closed walk and all closed walks generated subsequently that contain more than one cycle. The recursive application of the scan operation eventually decomposes all closed walks into cycles in finite number of steps. To see this, each time when “children” closed walks are generated by passing a “parent” closed walk through scan, the number of edges of the children closed walks must be smaller than that of the parent. The scan operation can be applied to each children which is not a cycle, which might generate more “grand-children” closed walks with even fewer edges. However, the recursive application of scan cannot continue indefinitely since all closed walks with two edges are cycles.

B Theorem 3 independent of choices of v_k 's

For E_{ij}^c , we claim that E_{ij}^c is independent of the choices of v_k 's. For any $1 \leq i \neq j \leq K$, (25) specifies that $(u, v) \in E_{ij}^c$ if and only if

$$d_{v_i s} + c_{st} + d_{tv_j} \geq d_{v_i u} + c_{uv} + d_{vv_i}, \quad \forall (s, t) \in E_{ij}. \quad (103)$$

However, by lemma 10.b and 10.c, (103) can be rewritten as

$$c_{st} \geq d_{sv_i} + d_{v_i u} + c_{uv} + d_{vv_j} + d_{v_j t} = d_{su} + c_{uv} + d_{vt}.$$

This shows that E_{ij}^c is independent of the choices of v_k 's.

Next, we establish that the set of R_0^* is independent of the choices of v_k 's. Let \tilde{G} be the condensation, where the representing nodes of the equivalence classes are v_1, \dots, v_K . Now instead, let $\hat{v}_1, \dots, \hat{v}_K$, where \hat{v}_k can be different from v_k , be the representing nodes. Let \hat{G} be the corresponding condensation. By Definition 8, (v_i, v_j) is an edge of \tilde{G} if and only if (\hat{v}_i, \hat{v}_j) is an edge of \hat{G} . In addition, it is claimed that if $\tilde{c}_{\hat{v}_i \hat{v}_j}$ is the edge weight of (\hat{v}_i, \hat{v}_j) in \hat{G} , then it holds that

$$\tilde{c}_{\hat{v}_i \hat{v}_j} = d_{\hat{v}_i v_i} + \tilde{c}_{v_i v_j} + d_{v_j \hat{v}_j}. \quad (104)$$

First, we claim that

$$\tilde{c}_{\hat{v}_i \hat{v}_j} \leq d_{\hat{v}_i v_i} + \tilde{c}_{v_i v_j} + d_{v_j \hat{v}_j}. \quad (105)$$

Note that, by (28) and Lemma 10.c, it holds that

$$\begin{aligned} d_{\hat{v}_i v_i} + \tilde{c}_{v_i v_j} + d_{v_j \hat{v}_j} &= d_{\hat{v}_i v_i} + d_{v_i v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v_j} + d_{v_j \hat{v}_j} \\ &= d_{\hat{v}_i v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j \hat{v}_j}. \end{aligned}$$

Hence, the right-hand side of (105) is in fact the weight of the walk $\hat{v}_i \rightsquigarrow v_{ij}^i \rightarrow v_{ij}^j \rightsquigarrow \hat{v}_j$. Consequently, by (27) (applied to \hat{G}), (105) holds. Next, to complete the proof of (104), suppose that

$$\tilde{c}_{\hat{v}_i \hat{v}_j} = d_{\hat{v}_i v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j \hat{v}_j} < d_{\hat{v}_i v_i} + \tilde{c}_{v_i v_j} + d_{v_j \hat{v}_j}. \quad (106)$$

Then, with Lemma 10.b and 10.c, (106) implies that

$$d_{v_i v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v_j} < \tilde{c}_{v_i v_j}.$$

This contradicts (27) in Definition 8. Consequently, (104) holds. With (104), it can be shown that for any sequence $\hat{v}_{k(1)}, \hat{v}_{k(2)}, \dots, \hat{v}_{k(m)}$, the inequality

$$\tilde{c}_{\hat{v}_{k(0)} \hat{v}_{k(m)}} \geq \tilde{c}_{\hat{v}_{k(0)} \hat{v}_{k(1)}} + \tilde{c}_{\hat{v}_{k(1)} \hat{v}_{k(2)}} + \dots + \tilde{c}_{\hat{v}_{k(m-1)} \hat{v}_{k(m)}} \quad (107)$$

is equivalent to

$$\begin{aligned} & d_{\hat{v}_{k(0)} v_{k(0)}} + \tilde{c}_{v_{k(0)} v_{k(m)}} + d_{v_{k(m)} \hat{v}_{k(m)}} \\ & \geq d_{\hat{v}_{k(0)} v_{k(0)}} + \tilde{c}_{v_{k(0)} v_{k(1)}} + d_{v_{k(1)} \hat{v}_{k(1)}} + d_{\hat{v}_{k(1)} v_{k(1)}} + \tilde{c}_{v_{k(1)} v_{k(2)}} + d_{v_{k(2)} \hat{v}_{k(2)}} \\ & \quad + \dots + d_{\hat{v}_{k(m-1)} v_{k(m-1)}} + c_{v_{k(m-1)} v_{k(m)}} + d_{v_{k(m)} \hat{v}_{k(m)}}. \end{aligned} \quad (108)$$

In (108), Lemma 10.b yields $d_{v_{k(q)}\hat{v}_{k(q)}} + d_{\hat{v}_{k(q)}v_{k(q)}} = 0$ for $q = 1, \dots, m-1$. Thus, (108) becomes

$$\tilde{c}_{v_{k(0)}v_{k(m)}} \geq \tilde{c}_{v_{k(0)}v_{k(1)}} + \tilde{c}_{v_{k(1)}v_{k(2)}} + \dots + \tilde{c}_{v_{k(m-1)}v_{k(m)}}. \quad (109)$$

That is, inequalities (107) and (109) are equivalent. The equivalence implies that \tilde{R} is a redundant edge set in \tilde{G} if and only if $\hat{R} := \{(\hat{v}_i, \hat{v}_j) \mid (v_i, v_j) \in \tilde{R} \iff (\hat{v}_i, \hat{v}_j) \in \hat{R}\}$ is a redundant edge set in \hat{G} with the same cardinality, and vice versa. Therefore, for $1 \leq i \neq j \leq K$, either there is an edge from $[v_i]$ to $[v_j]$ in the maximum redundant edge set of all condensations, or there is no edge from $[v_i]$ to $[v_j]$ in the maximum redundant edge set of any condensation. This is the second bullet of Remark 8. Consequently, the definition of R_{ij}^* (and hence R_0^*) in Theorem 3 is independent of the choices of v_k 's.

C Theorem 4 independent of arbitrary choices

The first bullet has been shown in Appendix B. The second bullet is shown as follows:

$$\begin{aligned} & \min_{(s,t) \in E_{ij}} d_{us} + c_{st} + d_{tv} \\ \stackrel{\text{Lemma 10.c}}{=} & \min_{(s,t) \in E_{ij}} d_{uv_i} + d_{v_i s} + c_{st} + d_{tv_j} + d_{v_j v} \\ = & d_{uv_i} + \left(\min_{(s,t) \in E_{ij}} d_{v_i s} + c_{st} + d_{tv_j} \right) + d_{v_j v} \\ \stackrel{\text{Definition 7.d}}{=} & d_{uv_i} + d_{v_i v_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v_j} + d_{v_j v} \\ \stackrel{\text{Lemma 10.c}}{=} & d_{uv_{ij}^i} + c_{v_{ij}^i v_{ij}^j} + d_{v_{ij}^j v}. \end{aligned}$$

References

- [1] D. M. Moyles and G. L. Thompson, “An algorithm for finding a minimum equivalent graph of a digraph,” *J. ACM*, vol. 16, no. 3, pp. 455–460, Jul. 1969. [Online]. Available: <http://doi.acm.org/10.1145/321526.321534>
- [2] A. Aho, M. Garey, and J. Ullman, “The transitive reduction of a directed graph,” *SIAM Journal on Computing*, vol. 1, no. 2, pp. 131–137, 1972.

- [3] J. Telgen, “Identifying redundant constraints and implicit equalities in systems of linear constraints,” *Management Science*, vol. 29, no. 10, pp. 1209–1222, 1983.
- [4] H. J. Greenberg, “Consistency, redundancy, and implied equalities in linear systems,” *Annals of Mathematics and Artificial Intelligence*, vol. 17, no. 1, pp. 37–83, 1996.
- [5] L. Finta and Z. Liu, “Single machine scheduling subject to precedence delays,” *Discrete Applied Mathematics*, vol. 70, no. 3, pp. 247–266, 1996.
- [6] E. Balas, J. K. Lenstra, and A. Vazacopoulos, “The one-machine problem with delayed precedence constraints and its use in job shop scheduling,” *Manage. Sci.*, vol. 41, no. 1, pp. 94–109, Jan. 1995.
- [7] C. Chekuri and R. Motwani, “Precedence constrained scheduling to minimize sum of weighted completion times on a single machine,” *Discrete Applied Mathematics*, vol. 98, no. 1–2, pp. 29–38, 1999.
- [8] P. Brucker, T. Hilbig, and J. Hurink, “A branch and bound algorithm for a single-machine scheduling problem with positive and negative time-lags,” *Discrete Applied Mathematics*, vol. 94, no. 1–3, pp. 77–99, 1999, proceedings of the Third International Conference on Graphs and Optimization GO-III.
- [9] E. D. Wikum, D. C. Llewellyn, and G. L. Nemhauser, “One-machine generalized precedence constrained scheduling problems,” *Operations Research Letters*, vol. 16, no. 2, pp. 87–99, 1994.
- [10] R. Möhring and M. Uetz, “Scheduling scarce resources in chemical engineering,” in *Mathematics – Key Technology for the Future*. Springer Berlin Heidelberg, 2003, pp. 637–650.
- [11] K. C. Sou, J. Weimer, H. Sandberg, and K.H. Johansson, “Scheduling smart home appliances using mixed integer linear programming,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, dec. 2011, pp. 5144 –5149.
- [12] K.C. Sou, M. Kördel, J. Wu, H. Sandberg, and K.H. Johansson, “Energy and co2 efficient scheduling of smart home appliances,” in *European Control Conference*, 2013.

- [13] P. Sucha and Z. Hanzalek, “Scheduling of tasks with precedence delays and relative deadlines framework for time-optimal dynamic reconfiguration of fpgas,” in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, April 2006.
- [14] J. M. Pinilla, J.-H. Kao, and F. Prinz, “Compact graph representation for solid freeform fabrication (sff),” *Journal of Manufacturing Systems*, vol. 19, no. 5, pp. 341 – 354, 2001.
- [15] K. C. Sou, H. Sandberg, and K. H. Johansson, “Nonserial dynamic programming with applications in smart home appliances scheduling – part II: Nonserial dynamic programming,” in *European Control Conference*, 2014.
- [16] J. Coffman, E.G. and R. Graham, “Optimal scheduling for two-processor systems,” *Acta Informatica*, vol. 1, no. 3, pp. 200–213, 1972.
- [17] J. Tsitsiklis and D. Bertsimas, *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [18] A. Schrijver, “A course in combinatorial optimization,” 2010, <http://homepages.cwi.nl/~lex/>.
- [19] K. C. Sou, H. Sandberg, and K. H. Johansson, “Nonserial dynamic programming with applications in smart home appliances scheduling – part I: Precedence graph simplification,” in *European Control Conference*, 2014.
- [20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.
- [21] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.
- [22] H. T. Hsu, “An algorithm for finding a minimal equivalent graph of a digraph,” *J. ACM*, vol. 22, no. 1, pp. 11–16, Jan. 1975.
- [23] S. Khuller, B. Raghavachari, and N. Young, “Approximating the minimum equivalent digraph,” *SIAM Journal on Computing*, vol. 24, no. 4, pp. 859–872, 1995.
- [24] R. Diestel, *Graph Theory*. Springer, 2012.

- [25] N. Biggs, *Discrete Mathematics*, 2nd ed. Oxford University Press, 2002.